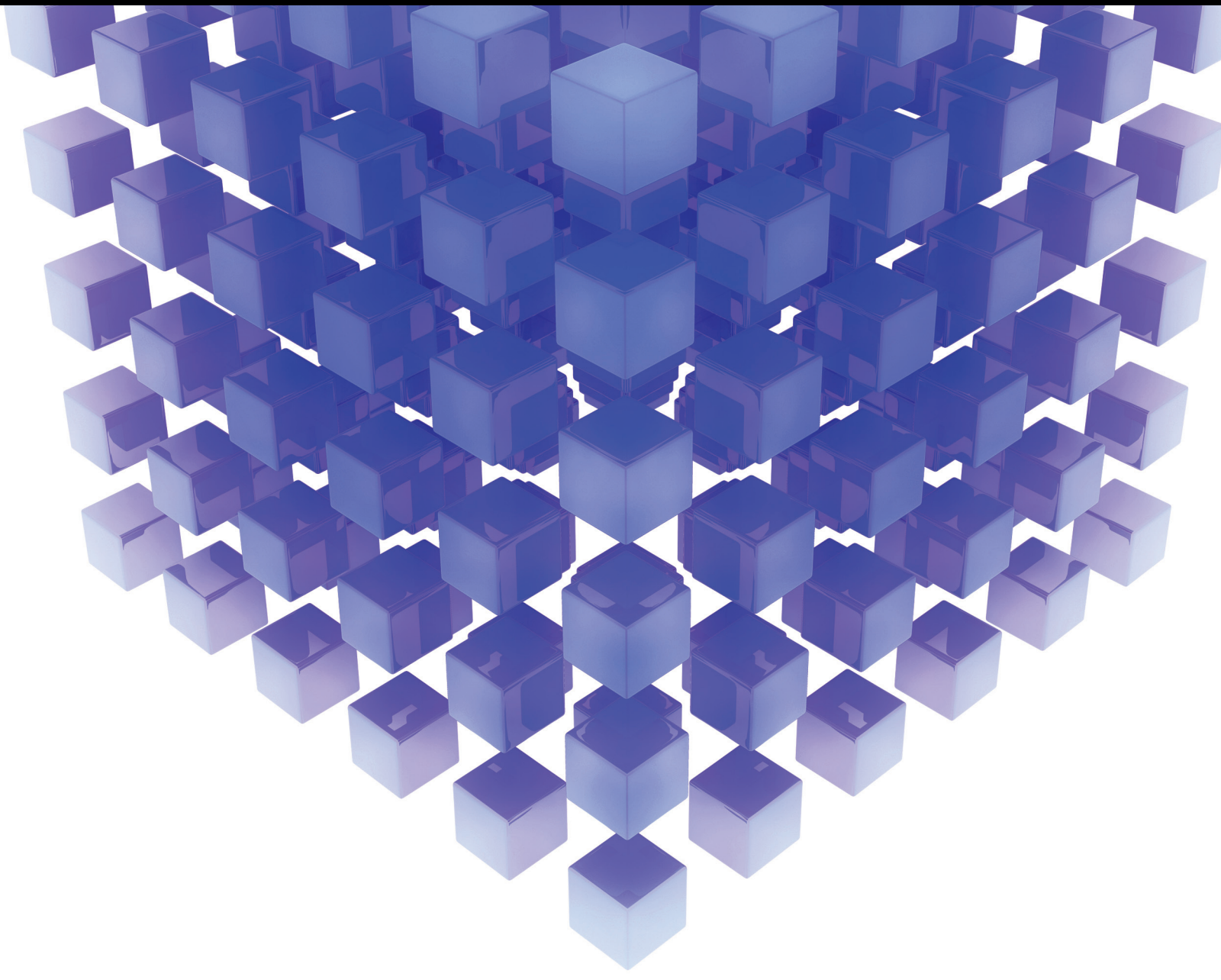
Mathematical Problems in Engineering

Computational Intelligence Approaches to Robotics, Automation, and Control

Guest Editors: Yi Chen, Yun Li, Huosheng Hu, Jun Zhang, Dongbing Gu, and Peter Xu



Computational Intelligence Approaches to Robotics, Automation, and Control

Mathematical Problems in Engineering

Computational Intelligence Approaches to Robotics, Automation, and Control

Guest Editors: Yi Chen, Yun Li, Huosheng Hu, Jun Zhang, Dongbing Gu, and Peter Xu

Editorial Board

Contents

Hindawi Publishing Corporation
Mathematical Problems in Engineering
Volume 2015, Article ID 620275, 1 page
http://dx.doi.org/10.1155/2015/620275

Editorial

Computational Intelligence Approaches to Robotics, Automation, and Control

Yi Chen,[1] Yun Li,[2] Huosheng Hu,[3] Jun Zhang,[4] Dongbing Gu,[3] and Peter Xu[5]

[1]*School of Engineering and Built Environment, Glasgow Caledonian University, Glasgow G4 0BA, UK*
[2]*School of Engineering, University of Glasgow, Glasgow G12 8QQ, UK*
[3]*School of Computer Science and Electronic Engineering, University of Essex, Wivenhoe Park, Colchester CO4 3SQ, UK*
[4]*Department of Computer Science, Sun Yat-Sen University, Guangzhou 510060, China*
[5]*Department of Mechanical Engineering, University of Auckland, Private Bag 92019, Auckland, New Zealand*

Correspondence should be addressed to Yi Chen; leo.chen.yi@live.co.uk

Received 11 September 2014; Accepted 11 September 2014

Computational intelligence (CI) approaches are nature-inspired methods, which offer a wealth of ideas for complex problems solving. Compared with the traditional approaches, the CI approaches are more powerful so that they do not need the reformulation of the problem to search a nonlinear and nondifferentiable space with real world conditions and need for massive parallelism. Another advantage of the CI is the flexibility of the fitness function formulation, which can be expressed as a proper function of the system's output and is suitable for multiobjective (MO) problems.

Robotics is a wide range research which includes design, construction, operation, and applications of robotic systems, as well as computer systems for control, sensory feedback, and information processing, in which CI approaches have been widely employed in automation and control that can take the place of humans in dangerous environments or manufacturing processes or resemble humans in appearance, behaviours, and cognition.

The special issue focuses on the theoretical, numerical, and experimental contributions that describe original research results, innovative concepts that address all aspects of robotics, and CI approaches and are applying their results in the context of robotics, automation, and control. The aim is to establish a common understanding about the state of the field and draw a road map on where the research is heading, highlight the issues, and discuss the possible solutions.

Twenty-three papers are accepted to this special issue after a thorough reviewing process; the acceptance rate is 38.33% of all the submissions.

The special issue reports the continuing efforts to understand the research and development of CI approaches for optimisations and field applications, academic excellence, and ground-breaking research results in robotics, automation, and control systems, which is dedicated to researchers who wish to enhance or broaden their knowledge and expertise on academic and industrial leadership in robotics, high potential growth global robotics market, for example, industrial robotics and professional/domestic service robotics in the European Union, United States, China, and Japan.

Acknowledgments

The authors would like to express their faithful gratitude to all the contributors of this special issue for their support and to all the reviewers for their constructive and timely comments.

Yi Chen
Yun Li
Huosheng Hu
Jun Zhang
Dongbing Gu
Peter Xu

Hindawi Publishing Corporation
Mathematical Problems in Engineering
Volume 2015, Article ID 814210, 18 pages
http://dx.doi.org/10.1155/2015/814210

Research Article

The Evolutionary Algorithm to Find Robust Pareto-Optimal Solutions over Time

Meirong Chen,[1,2] Yinan Guo,[1] Haiyuan Liu,[1,2] and Chun Wang[1]

[1] *School of Information and Electrical Engineering, China University of Mining and Technology, Xuzhou 221116, China*
[2] *Science College, China University of Mining and Technology, Xuzhou 221116, China*

Correspondence should be addressed to Yinan Guo; guoyinan@cumt.edu.cn

Received 7 March 2014; Revised 28 May 2014; Accepted 17 August 2014

Academic Editor: Yun Li

In dynamic multiobjective optimization problems, the environmental parameters change over time, which makes the true pareto fronts shifted. So far, most works of research on dynamic multiobjective optimization methods have concentrated on detecting the changed environment and triggering the population based optimization methods so as to track the moving pareto fronts over time. Yet, in many real-world applications, it is not necessary to find the optimal nondominant solutions in each dynamic environment. To solve this weakness, a novel method called robust pareto-optimal solution over time is proposed. It is in fact to replace the optimal pareto front at each time-varying moment with the series of robust pareto-optimal solutions. This means that each robust solution can fit for more than one time-varying moment. Two metrics, including the average survival time and average robust generational distance, are present to measure the robustness of the robust pareto solution set. Another contribution is to construct the algorithm framework searching for robust pareto-optimal solutions over time based on the survival time. Experimental results indicate that this definition is a more practical and time-saving method of addressing dynamic multiobjective optimization problems changing over time.

1. Introduction

In many practical fields, such as engineering design, scientific computing, social economy, and network communication, there exist a large number of complex optimization problems. Particularly, many optimization problems contain multiple objective functions and dynamic parameters that make the objective functions change over time. Moreover, the number of objective functions and constraints may also vary from time to time. We call multiobjective optimization problems with above uncertain factors as dynamic multiobjective optimization problems (DMOPs). In this paper, we focus on the DMOPs with continuously changed dynamic parameters. Suppose $F(\vec{x}, \vec{\alpha}_t) = \{f_1(\vec{x}, \vec{\alpha}_t), f_2(\vec{x}, \vec{\alpha}_t), \ldots, f_M(\vec{x}, \vec{\alpha}_t)\}$ is the objective vector in DMOPs and $\vec{\alpha}_t$ is the dynamic parameters depending on t. The aim of DMOPs is to find the pareto front (PF^t) approximating to the true pareto front of $F(\vec{x}, \vec{\alpha}_t)$ as soon as possible for all of the dynamic environment.

Aiming at tracking the moving true pareto fronts over time, dynamic multiobjective evolutionary optimization algorithms (DMOEA) were proposed [1–12]. The universal framework of DMOEA is presented in Algorithm 1. First of all, we need to accurately judge whether the environment has changed. It is the basic premise of using evolutionary optimization methods to respond to the environmental changes. The most common change-detection approach is to reevaluate the detectors. The detectors can be the current best solutions, a memory-based subpopulation, or a feasible subpopulation [5].

Once the various environment parameters happened, the new evolutionary optimization process was triggered. Many evolutionary algorithms with good performances on the static multiobjective optimization problems have been introduced into the DMOPs one after another, such as genetic algorithm [6], particle swarm algorithm [7], differential evolution algorithm [8], quantum immune clonal coevolutionary algorithm [9], and memetic algorithm [10]. As we know, the goal of the static optimization problems is to make the population gradually converge to optimal nondominant solution. The diversity of the population becomes weakened

```
Set t = 0;
Initialize a population P_t;
Repeat
    Detect the change of the environment;
    if the environment varies then
        Set t = 0;
        Reinitialize the population P_t;
    end
    Excute the evolutionary operations;
    t = t + 1;
    Until termination criteria met
```

Algorithm 1: DMOEA (the universal framework).

during the optimization methods. How to improve and maintain the evolutionary algorithm's ability to adapt to the various environment is a major challenge faced in the dynamic evolutionary computation. In recent years, various methods have been used to improve the diversity of the population. Chen et al. [11] extended additional objectives to deal with the DMOPs. Individual diversity is used as an additional objective to provide the historical information. Zhou et al. [12] proposed a population prediction strategy to improve the DMOEA's performance when the new environment is detected. The reinitial population was formed by a center point and a manifold. The track of center points was preserved to train autoregressive model so as to predict the center point in the new environment. A new coevolutionary paradigm [2] combining competition with cooperation was proposed to track the true pareto front in dynamic environment.

The conventional methods are mostly to trigger the multiobjective optimization process after detecting the change and then finish the evolution process as the following new environment occurs. Yet, this is impractical in many real-world optimization problems due to the following reasons. These methods are not suitable for rapidly changing environments, in which the environmental parameters vary quickly or frequently. Moreover, it is difficult to find the satisfied pareto fronts before detecting the change of fitness landscape because the methods are extremely time-consuming.

To address the above concern, several methods to find robust pareto front of multiobjective optimization problems with noise were presented by [13–17]. The emphasis is to seek an insensitive robust pareto front instead of the global optimal pareto front. The detailed definition about the robustness is illustrated in Section 2. For dynamic scalar optimization problems, robust optimization over time (ROOT) has been defined clearly by Yu et al. [18]. The task for ROOT is to find a sequence of robust solutions over time intervals. They have acceptable qualities and are relatively insensitive to the dynamic environment. By ROOT, the uncertainties in the parameter space and their cumulative effect on objective space are considered simultaneously. Furthermore, Jin et al. [19] gave a framework of ROOT which consists of a population-based optimization algorithm, the database, a fitness approximator, and a fitness predictor. A solution's robustness over time is estimated by both its past and its future performances. Subsequently, Fu et al. [20] provided a feasible algorithm in order to find robust optimal solutions over time and gave a detailed definition of the survival time and the average fitness. The robust solutions are expected to have longer survival time or larger average fitness. Though ROOT is easy to be realized and computed, it only fits for dynamic scalar optimization problems.

For DMOPs, the detailed robust definition over time is still an open issue. There were seldom relevant literatures about this area. We will introduce this idea about ROOT into DMOPs. The main contribution in this paper is to propose a novel concept on robust pareto-optima over time (RPOOT) to DMOPs that search for robust pareto-optimal solution set for all dynamic environments. Subsequently, the population-based multiobjective evolutionary algorithm is introduced to find RPOOT in terms of the nondomination solutions' robust performance. The robustness is measured by the survival time derived from the robust index given by Deb and Gupta [14]. We believe it is a more practical way of addressing continuously changed DMOPs.

The remainder of the paper is structured as follows. Section 2 presents a brief introduction to research on existing robust optimization methods and analyzes the existing problems in detail. Section 3 presents a class of DMOPs and describes formally the definition of RPOOT. In Section 4, the robustness and the performance metrics are defined for RPOOT. Furthermore, a population-based evolutionary algorithm to find robust pareto-optimal solutions over time is presented. Section 5 provides a brief overview of the existing benchmark functions for DMOPs and the experimental result for RPOOT. Conclusions and future work are presented in Section 6.

2. Related Works

In order to solve multiobjective optimization problems with uncontrollable variations, Li et al. [13] presented a robust multiobjective genetic algorithm by considering two objective functions: the fitness value f_v, which measures a solution's performance by a combined objective, and the robustness index η. They investigated the trade-off between the convergence and robustness of the nondominant solutions. Furthermore, Li proposed the Outer-Inner optimization structure. The outer subproblem was to simultaneously minimize the fitness value and to maximize the robustness index. The inner subproblem calculates the radius R, which represents a solution's robustness.

Deb and Gupta [14] extended an existing approach that finds robust solutions for single-objective optimization problems to MOPs with dynamic parameters. They defined the mean effective objective functions instead of the original objective functions.

Consider a multiobjective optimization problem as follows:

$$\begin{aligned} &\min && \mathbf{F}(\mathbf{x}) = \left(f_1(\mathbf{x}), f_2(\mathbf{x}), \ldots, f_M(\mathbf{x})\right) \\ &\text{subject to} && \mathbf{x} \in S. \end{aligned} \tag{1}$$

In order to avoid obtaining the global optimal solutions which are quite sensitive to such variable perturbation in their vicinity, the following two approaches are defined for robust optimization by Deb and Gupta [14].

Definition 1 (multiobjective robust solution of type I (MORS1)). A solution $\mathbf{x}^*$ is called a multiobjective robust solution of type I if it is the global feasible pareto-optimal solution to the following multiobjective minimization problem (suppose a δ-neighborhood of a solution $\mathbf{x}$ is $B_\delta(\mathbf{x})$):

$$\begin{aligned} &\text{Minimize} \quad \mathbf{F}^{\text{eff}}(\mathbf{x}) = \left(f_1^{\text{eff}}(\mathbf{x}), f_2^{\text{eff}}(\mathbf{x}), \ldots, f_M^{\text{eff}}(\mathbf{x})\right) \\ &\text{subject to} \quad \mathbf{x} \in S, \end{aligned} \tag{2}$$

where $f_j^{\text{eff}}(\mathbf{x})$ is defined as follows:

$$f_j^{\text{eff}}(\mathbf{x}) = \frac{1}{\left|B_\delta(\vec{x})\right|} \int_{\mathbf{y} \in B_\delta(\mathbf{x})} f_j(\mathbf{y})\, d\mathbf{y}. \tag{3}$$

Definition 2 (multiobjective robust solution of type II (MORS2)). A solution $\mathbf{x}^*$ is called a multiobjective robust solution of type II if it is the global feasible pareto-optimal solution to the following multiobjective minimization problem:

$$\begin{aligned} &\text{Minimize} \quad \mathbf{F}(\mathbf{x}) = \left(f_1(\mathbf{x}), f_2(\mathbf{x}), \ldots, f_M(\mathbf{x})\right), \\ &\text{subject to} \quad \frac{\left\|\mathbf{F}^{\text{eff}}(\mathbf{x}) - \mathbf{F}(\mathbf{x})\right\|}{\|\mathbf{F}(\mathbf{x})\|} \le \eta, \\ &\qquad\qquad \mathbf{x} \in S. \end{aligned} \tag{4}$$

MORS1 replaces the original objective function $\mathbf{F}(\mathbf{x})$ with the effective objective function $\mathbf{F}^{\text{eff}}(\mathbf{x})$. Where $\mathbf{f}_i^{\text{eff}}(\mathbf{x})$ is the mean of ith objective function values in the vicinity of $\mathbf{x}$. In Definition 2, the original objective functions need to be optimized. At the same time, the feasible solutions must satisfy the constraint; that is, the objective function values among neighboring solutions are limited to a user-defined threshold η. Subsequently, the solution's robustness is judged. Two kinds of performances can not be analyzed at the same time. Both definitions for robustness took the variable perturbation into account. However, this is not DMOPs in fact.

Furthermore, Barrico and Antunes [15] defined the degree of robustness based on the solutions' behavior in their neighborhood in the decision space. The degree of robustness was also used to evaluate the solutions' behavior in the neighborhood of the reference scenario in the space of the objective functions' coefficients [16]. The weakness of above definition about the degree of robust is that the pareto-optimal set has to be known in advance. To solve this problem, Cromvik et al. [17] put forward a new definition for robustness index and introduced the utility function to convert multiobjective optimization problem into an approximation for a single decision maker. In this method, the robust index cannot be used as an objective during the optimization.

In a word, the above methods can solve the multiobjective optimization problems with perturbation in the decision space or the space of the objective function coefficients. However, only a robust pareto front meeting all multiobjective optimization problems with the disturbance is found. In this paper, we will discuss the robust solution set to a class of DMOPs with changing parameters.

3. The Definition of RPOOT

In this paper, we focus on the DMOPs, in which the environmental parameters continuously change over time and keep stationary between two time-varying moments. Obviously, the true pareto fronts shift from time to time. In other words, the objective functions depending on t are deterministic during each changing stage. Hence, this kind of DMOPs with continuous parameters can be discretized into a series of multiobjective optimization problems (MOPs) at each time-varying moment. Namely, the pareto fronts at each time-varying moment are regarded as the basis of optimization.

Without loss of generality, dynamic multiobjective optimization problem is defined as follows:

$$\begin{aligned} &\min \quad F(\vec{x}, \vec{\alpha}_t) = \{f_1(\vec{x}, \vec{\alpha}_t), \\ &\qquad\qquad f_2(\vec{x}, \vec{\alpha}_t), \ldots, f_M(\vec{x}, \vec{\alpha}_t)\} \\ &\text{subject to} \quad \vec{x} \in S, \end{aligned} \tag{5}$$

where $\vec{x}$ stands for the decision variable vector, $S \subset R^N$ is the decision space, and R^M is the objective space. $F : (S, t) \to R^M$ consists of M objective functions $f_i(\vec{x}, t)$ ($i = 1, 2, \ldots, M$). $\vec{\alpha}_t$ is the time-depending parameter vector. $\bar{t} \in [0, t_{\text{end}}]$ represents the time. Suppose $1/T$ is the frequency for dynamic environments. There may be $N = \lfloor t_{\text{end}}/T \rfloor$ different time steps. DMOPs are divided into N MOPs, denoted by $\langle F(\vec{x}, \vec{\alpha}_1), F(\vec{x}, \vec{\alpha}_2), \ldots, F(\vec{x}, \vec{\alpha}_N)\rangle$.

Based on the definition of robust solutions' index given by Deb and Gupta [14] and the description of ROOT presented by Yu et al. [18], we propose a novel concept, called robust pareto-optimal over time (RPOOT) for DMOPs. It consists of a sequence of robust pareto-optimal sets, denoted by $\langle S_P^1, S_P^2, \ldots, S_P^l\rangle$ ($1 \le l \le N$). Each robust pareto-optimal solution S_P^i, $i = 1, 2, \ldots, l$ may not approximate to the true pareto front at each period between changes but is relatively closer to all true pareto fronts during at least two consecutive time intervals.

Two hypotheses for RPOOT are that the environmental parameters change over time with stationary periods T between changes and that the robust solution fits for the consecutive changes during $\bar{t} \in [t_l, t_u] \subseteq [0, t_{\text{end}}]$. With $l_\alpha = \lceil t_l/T \rceil$ and $u_\alpha = \lceil t_u/T \rceil$, we obtain the problem

$$\left\langle F\left(\vec{x}, \vec{\alpha}_{l_\alpha}\right), F\left(\vec{x}, \vec{\alpha}_{l_\alpha+1}\right), \ldots, F\left(\vec{x}, \vec{\alpha}_{u_\alpha}\right)\right\rangle, \tag{6}$$

where $\vec{\alpha}_i = \vec{\alpha}_{i-1} + \Delta\vec{\alpha}$. The dynamics parameter $\Delta\vec{\alpha}$ is a random variable obeying a certain distribution such as a Gaussian distribution or a Uniform distribution. Let $P(\Delta\vec{\alpha})$ be the probability density function of $\Delta\vec{\alpha}$. The nondominant solutions' performances are measured by the following two definitions.

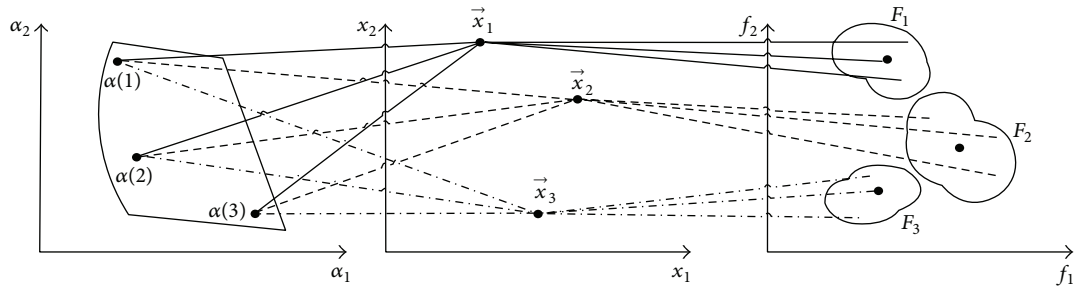


Figure 1: The relationship of neighborhoods among robust solutions in objective space (for 2-dimension spaces and all functions to be minimized).

Definition 3 (the average fitness). Consider

$$F(\vec{x}; l_\alpha, u_\alpha) = (f_1(\vec{x}; l_\alpha, u_\alpha), f_2(\vec{x}; l_\alpha, u_\alpha), \ldots, f_M^{\text{ave}}(\vec{x}; l_\alpha, u_\alpha)),$$

$$f_i(\vec{x}; l_\alpha, u_\alpha) = \frac{1}{u_\alpha - l_\alpha + 1} \sum_{j=1}^{u_\alpha - l_\alpha + 1} \int f_i\left(\vec{x}, \vec{\alpha}_{l_\alpha} + (j-1)\Delta\vec{\alpha}\right) p(\Delta\vec{\alpha})\, d\Delta\vec{\alpha}. \quad (7)$$

Definition 4 (the variance of fitness). Consider

$$D(\vec{x}; l_\alpha, u_\alpha) = (\delta_1(\vec{x}; l_\alpha, u_\alpha), \delta_2(\vec{x}; l_\alpha, u_\alpha), \ldots, \delta_M(\vec{x}; l_\alpha, u_\alpha)),$$

$$\delta_i(\vec{x}; l_\alpha, u_\alpha) = \frac{1}{u_\alpha - l_\alpha + 1} \cdot \sum_{j=1}^{u_\alpha - l_\alpha + 1} \int \left(f_i\left(\vec{x}; \vec{\alpha}_{l_\alpha} + (j-1)\Delta\vec{\alpha}\right) - f_i(\vec{x}; l_\alpha, u_\alpha)\right)^2 \cdot p(\Delta\vec{\alpha})\, d\Delta\vec{\alpha}. \quad (8)$$

In essence, F measures the average performance of each objective function within the time interval $[t_l, t_u]$. D is the degree of the performance influenced by changing the time-varying environment.

Definition 5 (robustness). A solution $\vec{x}_i(t)$ is called a robust pareto solution if it is nondominant individuals satisfied to the following DMOPs:

$$\text{Minimize} \quad F(\vec{x}; l_\alpha, u_\alpha) = (f_1(\vec{x}; l_\alpha, u_\alpha), f_2(\vec{x}; l_\alpha, u_\alpha), \ldots, f_M(\vec{x}; l_\alpha, u_\alpha))$$

$$\text{subject to} \quad \max_{1 \le j \le M} \delta_j(\vec{x}_i; l_\alpha, u_\alpha) < \eta. \quad (9)$$

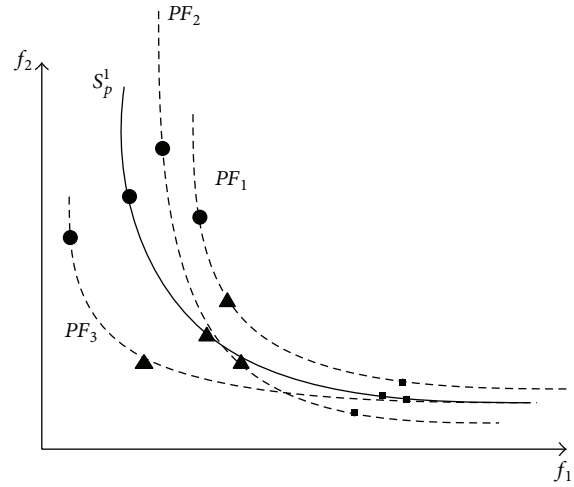


Figure 2: The definition of robust pareto-optimal set in objective space.

In fact, all solutions composed of robust pareto-optimal solution sets must be compromise. Subsequently, the nondominant relationship is defined based on the expected fitness vector over time. Define $\vec{x}_i(t) \succ \vec{x}_j(t)$ if $\forall l$, $f_l(\vec{x}_i; l_\alpha, u_\alpha) \le f_l(\vec{x}_j; l_\alpha, u_\alpha)$ and $\exists k$, $f_k(\vec{x}_i; l_\alpha, u_\alpha) < f_k(\vec{x}_j; l_\alpha, u_\alpha)$ is satisfied. $\vec{x}_i(t)$ and $\vec{x}_j(t)$ are compromise, if $\exists l$, $f_l(\vec{x}_i; l_\alpha, u_\alpha) > f_l(\vec{x}_j; l_\alpha, u_\alpha)$ and $\forall k \ne l$, $f_k(\vec{x}_i; l_\alpha, u_\alpha) \le f_k(\vec{x}_j; l_\alpha, u_\alpha)$. Consequently, we define robust pareto-optimal solutions as $S_P^t = \{\vec{x}^*(t) \mid \neg\exists \vec{x}_i(t) \succ \vec{x}^*(t))$ and $\max_{1 \le j \le M} \delta_j(\vec{x}_i; l_\alpha, u_\alpha) < \eta\}$. Obviously, RPOOT takes both the approximation and the robustness into account.

In Figure 1, the parameters vary in different periods, which map the uncertain area in the objective space. Obviously, $\vec{x}_3(t) \succ \vec{x}_2(t)$ and $\vec{x}_3(t) \succ \vec{x}_1(t)$. $\vec{x}_1(t)$ and $\vec{x}_2(t)$ are nondominated. However, $\vec{x}_2(t)$ is not robust due to its large variance. As shown in Figure 2, the true pareto-optimal sets for three consecutive time-varying periods are PF_1, PF_2, PF_3 plotted by the dash lines. S_p^1 plotted by real line in Figure 2 is the robust pareto-optimal solution. Obviously, S_p^1 is not the best pareto-optimal set during this period but the satisfied pareto front over time.

4. The Robust Solutions' Performance

In this paper, a robust pareto-optimal solution set needs not only to approximate to the true pareto front during each stage to the largest extent but also to satisfy the requirements of robustness. We would like to make a clear distinction between the definition of robustness for solutions and the approximation of each solution.

4.1. The Robustness. For DMOPs, the task of the traditional optimization methods is to find the pareto-optimal solutions after detecting the new environment. This is time-consuming. If the pareto-optimal front can meet the requirements of more than one kind of environment in a certain accuracy, the cost for search will be less. Consequently, the robustness has two means. One is the insensitivity to the fluctuation parameters. Deb and Gupta [14] has presented a general definition based on δ-neighborhood perturbation. The other is the survival time, which reflects how many consecutive changed environments this solution can fit for. Based on the above two aspects, corresponding metric called the survival time is proposed to measure the robustness in DMOPs.

Suppose $\vec{x}(t)$ is a nondomination solution at time t. The robustness of $\vec{x}(t)$ is defined by maximum survival time Lstarting from time t when all fitness values of $\vec{x}(t)$ from t to $t + L$ belong to η-neighborhood of $F(\vec{x}(t), \alpha_t)$.

Define the survival time of solution's robustness as L;

$$L\left(S_P^t, \eta\right) = \min_{\vec{x}(t)\in S_P^t} L^s\left(\vec{x}\left(t\right), \eta\right), \tag{10}$$

$$L^s\left(\vec{x}\left(t\right), \eta\right) = \max\left\{ l \left| \frac{\left\| \widehat{F}\left(\vec{x}\left(t\right), \vec{\alpha}_{t+i}\right) - F\left(\vec{x}\left(t\right), \vec{\alpha}_t\right) \right\|}{\left\| F\left(\vec{x}\left(t\right), \vec{\alpha}_t\right) \right\|} \le \eta, \forall i, 1 \le i \le l \right. \right\}. \tag{11}$$

$L^s(\vec{x}(t), \eta)$ stands for the survival time of any optimal solution $\vec{x}(t)$ in robust pareto-optimal set S_P^i. η is the user-defined threshold, which is the key parameter having a direct impact on L. The larger η means the tolerance to the varieties of the fitness is better. It makes L larger. $\|\cdot\|$ operator is used to measure the distance between the current fitness values and the future predictive fitness values. Here, the Euclidean norm is used. But any other suitable norm can also be adopted. $\widehat{F}(\vec{x}, \alpha_{t+i})$ is the approximated fitness instead of the real fitness value by a predictor [18]. Because the time-depending parameters vary randomly, each nondominated solution's real fitness values for each time-varying environment cannot be all evaluated accurately. So we need a prediction method to approximate the fitness in the future dynamic environments. As shown in Figure 3, if the robust pareto-optimal solutions at time t are satisfied during the consecutive time-varying moments from t to $t + L$, the fitnesses $F(\vec{x}, \vec{\alpha}_{t+1}), F(\vec{x}, \vec{\alpha}_{t+2}), \ldots, F(\vec{x}, \vec{\alpha}_{t+L})$ are all constricted to the η-neighborhood of $F(\vec{x}, \vec{\alpha}_t)$.

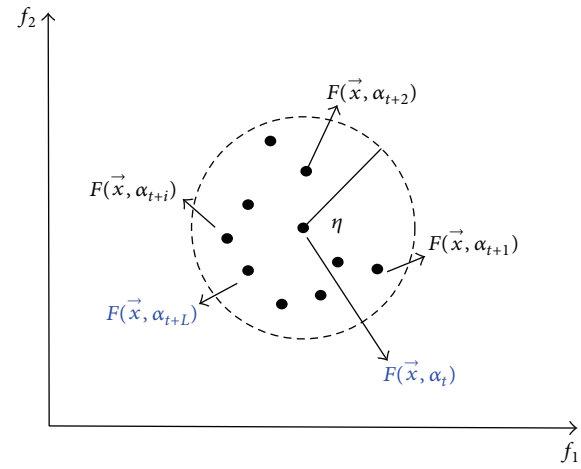


Figure 3: The definition of robust survival time in the neighborhood.

4.2. The Average Fitness Function. Generally, the integral part of (7) is not easy to be calculated since there is little knowledge to get the accurate probability density function. So we take the following average performance as the robustness of solution over the considered time interval.

We define S_P^t as a robust pareto-optimal solution over time if it is the global feasible pareto-optimal solution to the following multiobjective optimization problem:

$$\begin{aligned} \min \; F^{\text{ave}}\left(\vec{x}, \vec{\alpha}_t\right) &= \{f_1^{\text{ave}}\left(\vec{x}, \vec{\alpha}_t\right), \\ &\quad f_2^{\text{ave}}\left(\vec{x}, \vec{\alpha}_t\right), \ldots, f_M^{\text{ave}}\left(\vec{x}, \vec{\alpha}_t\right)\} \\ f_i^{\text{ave}}\left(\vec{x}, \vec{\alpha}_t\right) &= \frac{1}{L+1}\sum_{j=0}^{L} \widehat{f}_i\left(\vec{x}, \vec{\alpha}_{(t+j)}\right), \end{aligned} \tag{12}$$

where $f_i^{\text{ave}}(\vec{x}, \vec{\alpha}_t)$ stands for the average fitness value of ith objective during the consecutive time-varying moments from t to $t + L$. Subsequently, $F^{\text{ave}}(\vec{x}, \vec{\alpha}_t)$ measures the average performance of S_P^t in each objective during the time interval L. The smaller average fitness value means that S_P^t is more approximate to the true pareto fronts of corresponding time-varying environments.

In a word, the average survival time reflects the robustness of S_P^t on the time space; that is, how many time-varying environments it fits for. The average fitness value measures how the pareto-optimal fronts approximate to the true pareto fronts in the objective space during these dynamic stages. Consequently, a robust pareto-optimal solution must be the one with the minimum average fitness values.

4.3. The Population-Based Optimization Method. Taking above metrics as the objectives, a novel framework to solve RPOOT problems is constructed for DMOPs. In the time-varying moments, many population-based evolutionary algorithms may be adopted to find a set of robust pareto-optimal fronts over time, denoted by $S_P^1, S_P^2, \ldots, S_P^l$.

It is worth noting that we must take into account the future performance of all objectives during the calculation of

Set $t = 1$;
Set evolutionary generation $g = 0$;
Initialize a population P_t;
Evaluate the fitness value and survival time L of every individual in P_t;
Caculate the average fitness value F^{ave} by formula (12);
Repeat
  $t_0 = t$;
  Detect change in the environment landspaces;
  if the environment changes **then**
    Set $t = t + 1$;
  end
  if $t = t_0 + L$
    Reinitialize the population P_t;
    Evaluate the fitness value and survival time L of every individual in P_t;
    Caculate the average fitness value F^{ave};
    Excute evolutionary operations;
  else
    Optimize the t_0th MOP by MOEA/D;
  End
    $g = g + 1$;
Until termination criteria met

Algorithm 2: EA for RPOOT (a framework finding RPOOT).

robust survival time and robustness performance. We assume that future performance can be estimated by a database and a predictor. The database is used to store historical data, and the task of the predictor is to estimate a solution's future performances [19]. A framework of population-based optimization algorithm for RPOOT is presented as following Algorithm 2. By this method, the robust pareto solution could not only approximate to the true pareto front as close as possible but also fit for more than one dynamic environment.

4.4. The Measurement of Algorithm Performance. It is important to measure the performance of the RPOOT algorithm. We should not only consider the robustness of the robust pareto-optimal solutions but also consider the accuracy of the solutions to the true pareto front. On the one hand, the average survival time measures the solutions' robustness on the time scale. On the other hand, the average robust metric measures the approximation to the true pareto front in the objective space.

4.4.1. The Average Survival Time. In DMOPs, the robustness of the robust optimal pareto solutions is measured by the average survival time. Moreover, the algorithms will be compared across the whole time period $t = 1, 2, \ldots, N$. Based on the robustness defined in (11), the robustness of the algorithm performance is defined as follows:

$$\overline{L} = \frac{1}{N}\sum_{t=1}^{N} L\left(S_p^t, \eta\right). \tag{13}$$

$L(S_p^t, \eta)$ is the robustness survival time of the nondomination solutions obtained from the algorithm during the timeline. Obviously, the longer the average survival time is, the better the robustness of solutions is. The robustness of optimal parato front on the time scale can be reflected by the average survival time. Moreover, L depends on the threshold η. Therefore, a more exhaustive analysis is necessary for the robust optimal pareto solutions under different η.

4.4.2. The Average Robust Generational Distance. This metric reflects the quality of the robust nondominant solution sets. The general distance (GD) [21] indicates how close the obtained PFs are to the true pareto front in multiobjective optimization problems. Furthermore, the inverted generational distance (IGD) [21] is used to assess the approximation performance of the algorithms. IGD measures both the diversity and the convergence of the population. In our experimental studies, robust generational distance (RGD) and robust inverted generational distance (RIGD) indicate the average distance between each robust optimal pareto front and the true pareto fronts within its survival time. Suppose l is the size of robust nondominant pareto solution set during the whole time interval. The RGD and RIGD metrics are defined as follows:

$$\begin{aligned} \text{RGD}^{\text{RPOOT}} &= \frac{1}{l}\sum_{i=1}^{l}\text{MGD}\,(i), \\ \text{MGD}\,(i) &= \max_{k=t_i,\ldots,t_i+L_{t_i}} \text{GD}\,(k), \end{aligned} \tag{14}$$

$$\begin{aligned} \text{RIGD}^{\text{RPOOT}} &= \frac{1}{l}\sum_{i=1}^{l}\text{MIGD}\,(i), \\ \text{MIGD}\,(i) &= \max_{k=t_i,\ldots,t_i+L_{t_i}} \text{IGD}\,(k), \end{aligned} \tag{15}$$

where t_i stands for the time step of robust nondominant solution set. L_{t_i} is their survival time. $\mathrm{GD}(k) = (1/|P^k|)\sum_{v\in P^k} d(PF^k, v))$ and $\mathrm{IGD}(k) = (1/|PF^k|)\sum_{v\in PF^k} d(v, P^k)$. PF^k is a set of uniformly distributed optimal solutions in the true PF at t; P^t is the solutions obtained at t. $d(PF^k, v) = \min_{u\in PF^k}\sqrt{\sum_{j=1}^{m}(f_j^{(u)} - f_j^{(v)})^2}$ is the distance between v and PF^k. $d(v, P^k) = \min_{u\in P^k}\sqrt{\sum_{j=1}^{m}(f_j^{(v)} - f_j^{(u)})^2}$ is the distance between v and P^k. $|P^k|$ and $|PF^k|$ are the cardinalities of P^k and PF^k. In our experiments, we select 100 evenly distributed solutions in *PFs*.

5. Analysis of the Experimental Results

In this section, eight dynamic multiobjective benchmark functions are adopted in the experiments. Simulation results and further analysis on solutions' performance are conducted in Section 5.2.

5.1. Benchmark Functions. Eight dynamic multiobjective benchmark functions are adopted to test whether or not the algorithm can find robust pareto-optimal solutions set. The first five functions are FDA1-FDA5 presented by Farina et al. [3]. The other three functions are DMOP1, DMOP2, and DMOP3 [22]. FDA4 and FDA5 contain three objectives, and the others include two objectives. τ is the generation counter. τ_t is the number of iterations under the time window t. n_t is the number of distinct steps under t, which controls the distance between two consecutive PSs.

The first type of benchmark are FDA1, FDA4, and DMOP3. For Type I problem, only the pareto sets (PSs) in the decision space dynamically change over time. However, the corresponding pareto fronts (PFs) in the objective space do not change with time. At any moment, the optimal pareto fronts are respectively $f_2 = 1 - \sqrt{f_1}$, $f_1^2 + f_2^2 + f_3^2 = 1$, and $f_2 = 1 - \sqrt{f_1}$. FDA2 and DMOP1 belong to Type III problems, in which only the PFs in the objective space change while the PSs in the decision space remain the same. The optimal pareto front of FDA2 is $f_2 = 1 - f_1^{(0.75+0.7\sin(0.5\pi t))^{-1}}$ that changes from a convex to a nonconvex shape. DMOP1 has a convex optimal pareto front $f_2 = 1 - f_1^{(1.25+0.75\sin(0.5\pi t))}$. FDA3, FDA5, and DMOP2 belonging to Type II problems have changing PSs and PFs: $f_2 = 1 - f_1^{(0.75+0.7\sin(0.5\pi t))^{-1}}$, $f_1^2 + f_2^2 + f_3^2 = (1 + G(t))^2$, $f_2 = 1 - f_1^{(1.25+0.75\sin(0.5\pi t))}$. The definitions of these dynamic multiobjective benchmark functions are summarized in Table 1. Their true PFs when $t = 5, 10, 20, 23, 26, 34$ are shown in Figure 4.

In this paper, we adopt a multiobjective evolutionary algorithm based on decomposition (MOEA/D) [21] to track the moving optimal pareto front over time. In MOEA/D, the penalty-based boundary intersection (PBI) approach is used as the surrogate model. For the benchmark functions, the population size is 100. In all experiments, time-varying moment t alters with the evolutionary generation and is associated with the parameter τ_t. The larger τ_t means that the environmental parameters change more infrequently and multiobjective optimization algorithms can spend more sufficient time tracking the new pareto front. Otherwise, the less τ_t makes multiobjective optimization algorithm triggered after detecting the new environment, hardly finding the satisfied pareto solutions closed to the true pareto front during the limited iterations. Now we discuss the algorithm performances under different frequencies. As shown in Figure 5, when the environment changes every 20 generations, the algorithm can track dynamic pareto-optimal fronts better. But within 5 generations, the satisfied PFs for the new environment are difficultly obtained. As a result, for all experiments, $\tau_t = 20$. The corresponding optimal pareto fronts of each benchmark function gotten by MOEA/D are shown below in Figure 6.

5.2. Simulation Results and Analysis. In this section, two groups of experiments have been done. In the first group, the optimal nondomination solutions starting from the time t_0 are obtained. If the minimum survival time of this moment is L, robust pareto solution fits for the environment from t_0 to $t_0 + L$. The process is repeated until the last environment occurs. The purpose of the second group of experiments is to obtain each robust optimal pareto front, respectively, for RPOOT at 100 time-varying moments.

5.2.1. The Effect of Neighborhood Size η. The neighborhood size η directly influences the evaluation criterion for the robust optimal pareto fronts. In the first group of experiments, we compare and analyze the performances of the robust optimal pareto fronts under different thresholds η. The numbers of robust pareto fronts (NRPFs) on 100 time-varying moments are listed in Table 2. The statistical average results of RGD and RIGD on eight benchmarks over 15 runs also can be found in Table 3 under different neighborhood sizes η.

As shown in Tables 2 and 3, with the increasing of the neighborhood size η, less robust pareto fronts are contained in the whole period. It means that the robustness of RPOOT is better. At the same time, their average inverted generational distances become larger, which means that the convergence of the RPOOT is worse. For each benchmark function, the standard errors of NRPF, RGD, and RIGD over 15 runs are given in Tables 2 and 3. The means and standard errors shown in these tables indicate that the stability of RPOOTs is good enough. We choose $\eta = 0.4$ in the following experiments. The robust optimal pareto fronts obtained from MOEA/D are shown in Figure 7.

From Figures 7(a)–7(h) and Table 4, we find that the GD and IGD of tracking multiobjective (TMO) algorithm are less than RGD and RIGD of RPOOT algorithm. But the number of robust pareto solution sets of RPOOT is far less than 100.

5.2.2. The Average Survival Time. The second group of experiments record each robust pareto front in RPOOT at each time-varying moment. Independent 15-time run is done for each benchmark. The results of the second group of experiments are plotted in Figure 8. It can be seen from Figure 8 that the average survival time of the robust pareto

Table 1: The dynamic benchmark functions.

Benchmark function	Definition	Type
FDA1	$f_1(X_{\mathrm{I}}) = x_1$ $f_2 = g \cdot h$ $g(X_{\mathrm{II}}) = 1 + \sum_{x_i \in X_{\mathrm{II}}} (x_i - G(t))^2$ $h(f_1, g) = 1 - \sqrt{\frac{f_1}{g}}$ $G(t) = \sin(0.5\pi t), \quad t = \frac{1}{n_t}\left\lfloor \frac{\tau}{\tau_t} \right\rfloor$ where: $m = 10, X_{\mathrm{I}} = (x_1) \in [0,1]; \quad X_{\mathrm{I}} = (x_2, \ldots, x_m) \in [-1,1]$	Type I
FDA2	$f_1(X_{\mathrm{I}}) = x_1$ $f_2 = g \cdot h$ $g(X_{\mathrm{II}}) = 1 + \sum_{x_i \in X_{\mathrm{II}}} x_i^2$ $h(f_1, g) = 1 - \left(\frac{f_1}{g}\right)^{H(t)^{-1}}$ $H(t) = 0.75 + 0.7\sin(0.5\pi t), \quad t = \frac{1}{n_t}\left\lfloor \frac{\tau}{\tau_t} \right\rfloor$ where: $X_{\mathrm{I}} = (x_1) \in [0,1]; \quad X_{\mathrm{II}} \in [-1,1], \lvert X_{\mathrm{II}} \rvert = 15$	Type III
FDA3	$f_1(X_{\mathrm{I}}) = \sum_{x_i \in X_{\mathrm{I}}}^{n} x_i^{F(t)}$ $f_2 = g \cdot h$ $g(X_{\mathrm{II}}) = 1 + G(t) + \sum_{x_i \in X_{\mathrm{II}}} (x_i - G(t))^2$ $h(f_1, g) = 1 - \sqrt{\frac{f_1}{g}}$ $G(t) = \lvert \sin(0.5\pi t) \rvert$ $F(t) = 10^{2\sin(0.5\pi t)}, \quad t = \frac{1}{n_t}\left\lfloor \frac{\tau}{\tau_t} \right\rfloor$ where: $\lvert X_{\mathrm{I}} \rvert = 1, \lvert X_{\mathrm{II}} \rvert = 9, X_{\mathrm{I}} \in [0,1]; X_{\mathrm{II}} \in [-1,1]$	Type II
FDA4	$f_1(x) = (1+g)\cos(0.5\pi x_1)\cos(0.5\pi x_2)$ $f_2(x) = (1+g)\cos(0.5\pi x_1)\sin(0.5\pi x_2)$ $f_3(x) = (1+g)\sin(0.5\pi x_1)$ $g(x) = \sum_{i=3}^{n}(x_i - G(t))^2$ $G(t) = \lvert \sin(0.5\pi t) \rvert, \quad t = \frac{1}{n_t}\left\lfloor \frac{\tau}{\tau_t} \right\rfloor$ where: $x_i \in [0,1], n = 12$	Type I
FDA5	$f_1(x) = (1+g)\cos(0.5\pi y_1)\cos(0.5\pi y_2)$ $f_2(x) = (1+g)\cos(0.5\pi y_1)\sin(0.5\pi y_2)$ $f_3(x) = (1+g)\sin(0.5\pi y_1)$ $g(x) = G(t) + \sum_{i=3}^{n}(x_i - G(t))^2, \quad y_i = x_i^{F(t)}$ $G(t) = \lvert \sin(0.5\pi t) \rvert, \quad F(t) = 1 + 100\sin^4(0.5\pi t), \quad t = \frac{1}{n_t}\left\lfloor \frac{\tau}{\tau_t} \right\rfloor$ where: $x_i \in [0,1], n = 12$	Type II
DMOP1	$f_1(x_1) = x_1$ $f_2 = g \cdot h$ $g(x_2, \ldots, x_m, t) = 1 + 9\sum_{i=2}^{m} x_i^2$ $h(f_1, g) = 1 - \left(\frac{f_1}{g}\right)^{H(t)}$ $H(t) = 0.75\sin(0.5\pi t) + 1.25, \quad t = \frac{1}{n_t}\left\lfloor \frac{\tau}{\tau_t} \right\rfloor$ where: $m = 10, \ x_i \in [0,1], \forall i = 1, 2, \ldots, m$	Type III

Table 1: Continued.

Benchmark function	Definition	Type
DMOP2	$f_1(x_1) = x_1$ $f_2 = g \cdot h$ $g(x_2, \ldots, x_m, t) = 1 + \sum_{i=2}^{m} (x_i - G(t))^{H(t)}$ $h(f_1, g) = 1 - \left(\frac{f_1}{g}\right)^{H(t)}$ $H(t) = 0.75 \sin(0.5\pi t) + 1.25, \quad t = \frac{1}{n_t}\left\lfloor \frac{\tau}{\tau_t} \right\rfloor$ where: $m = 10,\ x_i \in [0,1], \forall i = 1, 2, \ldots, m$	Type II
DMOP3	$f_1(X) = x_1$ $f_2 = g \cdot h$ $g(X, t) = 1 + 9\sum_{i=2}^{n} (x_i - G(t))^2$ $h(X, t) = 1 - \sqrt{\frac{f_1}{g}}$ $G(t) = \sin(0.5\pi t), \quad t = \frac{1}{n_t}\left\lfloor \frac{\tau}{\tau_t} \right\rfloor$ where: $X \in [0,1] \times [-1,2]^{n-1}, m = 10$	Type II

Table 2: Comparison of the number of robust pareto fronts (NRPFs) on 100 time-varying moments under different η.

Functions	Measures		η							
			$\eta = 0.1$	$\eta = 0.2$	$\eta = 0.3$	$\eta = 0.4$	$\eta = 0.5$	$\eta = 0.6$	$\eta = 0.7$	$\eta = 0.8$
FDA1	NRPFs	μ	70.9	61	50	37.4	35.9	30	33.1	30
		σ	0.3162	0	0	0.5164	0.3162	0	0.7379	0
FDA2	NRPFs	μ	40	19	9	9.4	1	1.8	2	1.6
		σ	0	0	0	1.2649	0	0.4216	0	0.5164
FDA3	NRPFs	μ	83.4	69	64	51.2	45.5	35.5	34.7	30.8
		σ	0.5164	0	0	0.6325	1.5811	0.9718	0.4830	0.6325
FDA4	NRPFs	μ	79	69.5	56	42.4	36.1	33.4	28.9	29.9
		σ	0	1.5811	0	1.2649	1.4491	1.3499	0.3162	0.3162
FDA5	NRPFs	μ	90	90	70	58.5	50	61.7	45	35
		σ	0	0	0	1.0801	0	0.4830	0	0
DMOP1	NRPFs	μ	24.8	5.9	6	1	1	1	1	1
		σ	0.6325	1.4491	0	0	0	0	0	0
DMOP2	NRPFs	μ	76.4	68	54.5	39.3	32.3	32.8	29.2	28
		σ	1.2649	0	0.7071	1.1595	0.9487	0.6325	0.7888	0
DMOP3	NRPFs	μ	90	89.6	84.6	78.8	80	80	80	79.8
		σ	0	0.5164	0.6992	1.0328	0	0	0	0.6325

front at each time-varying moment is far larger than 1. The results of the average survival time listed in Table 5 are all far longer than changing time 100.

6. Conclusions

Dynamic multiobjective optimization problems with changing parameters widely exist in real life. The aim of the traditional optimization algorithms is to track the optimal pareto solution set after detecting the environment change efficiently. These algorithms may not obtain the satisfied nondominant solutions between two time-varying moments. In this paper, we proposed a new perspective for solving DMOPs with consecutive time-varying periods. Its goal is to find the robust pareto solution set over time. Three contributions are contained in RPOOT. At first, the detailed concept

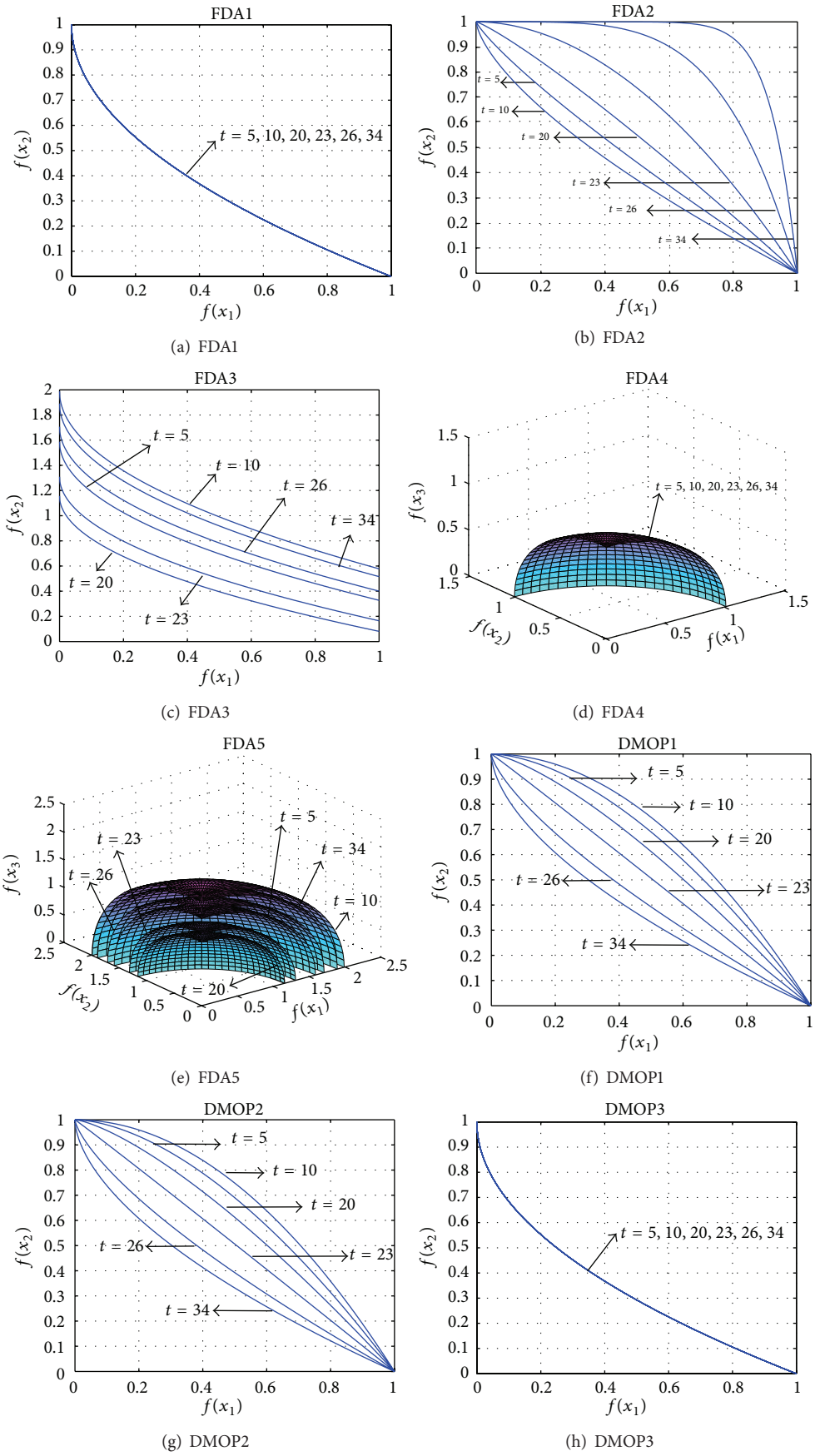
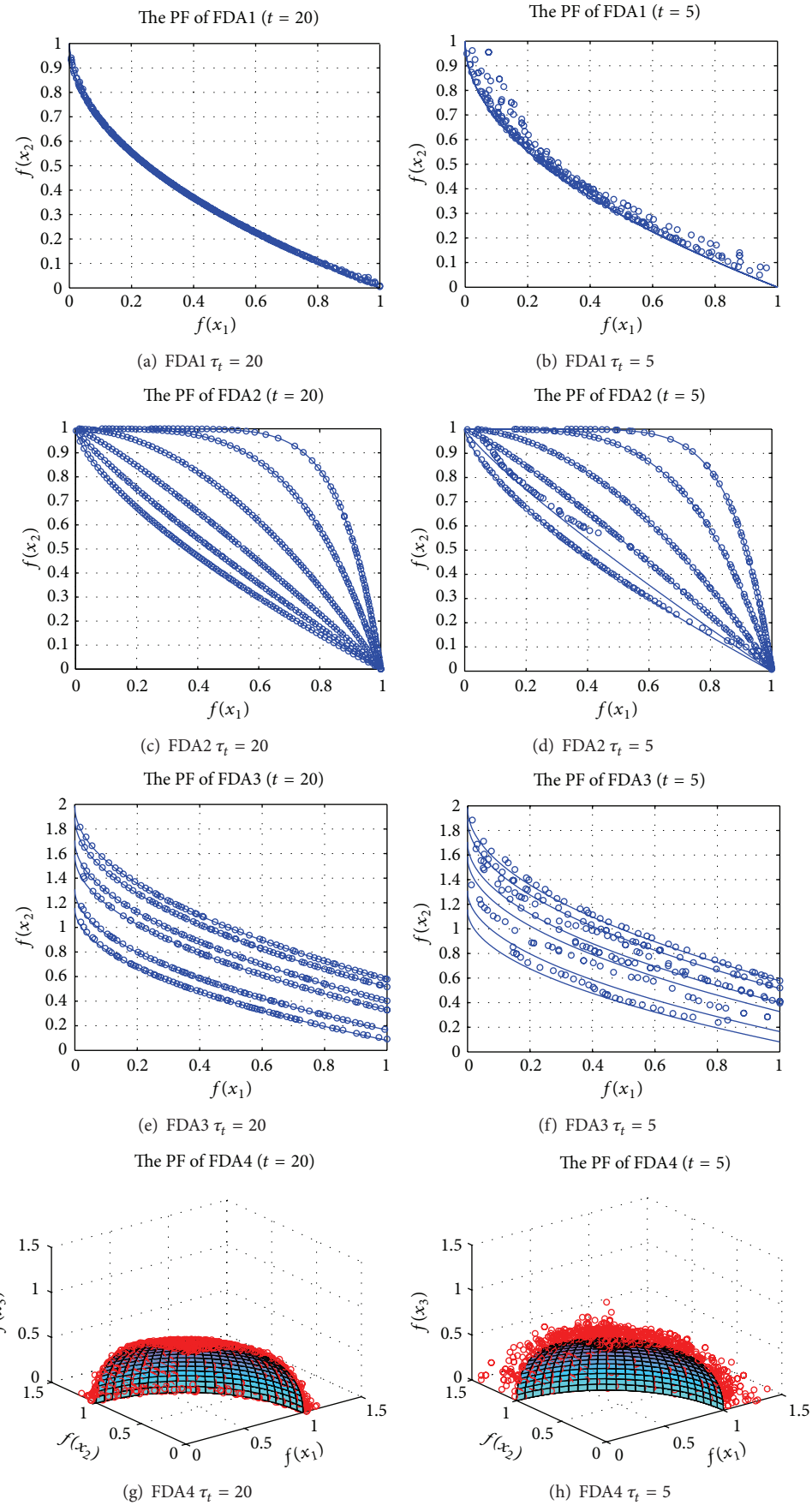


Figure 4: The true pareto fronts of benchmark functions.



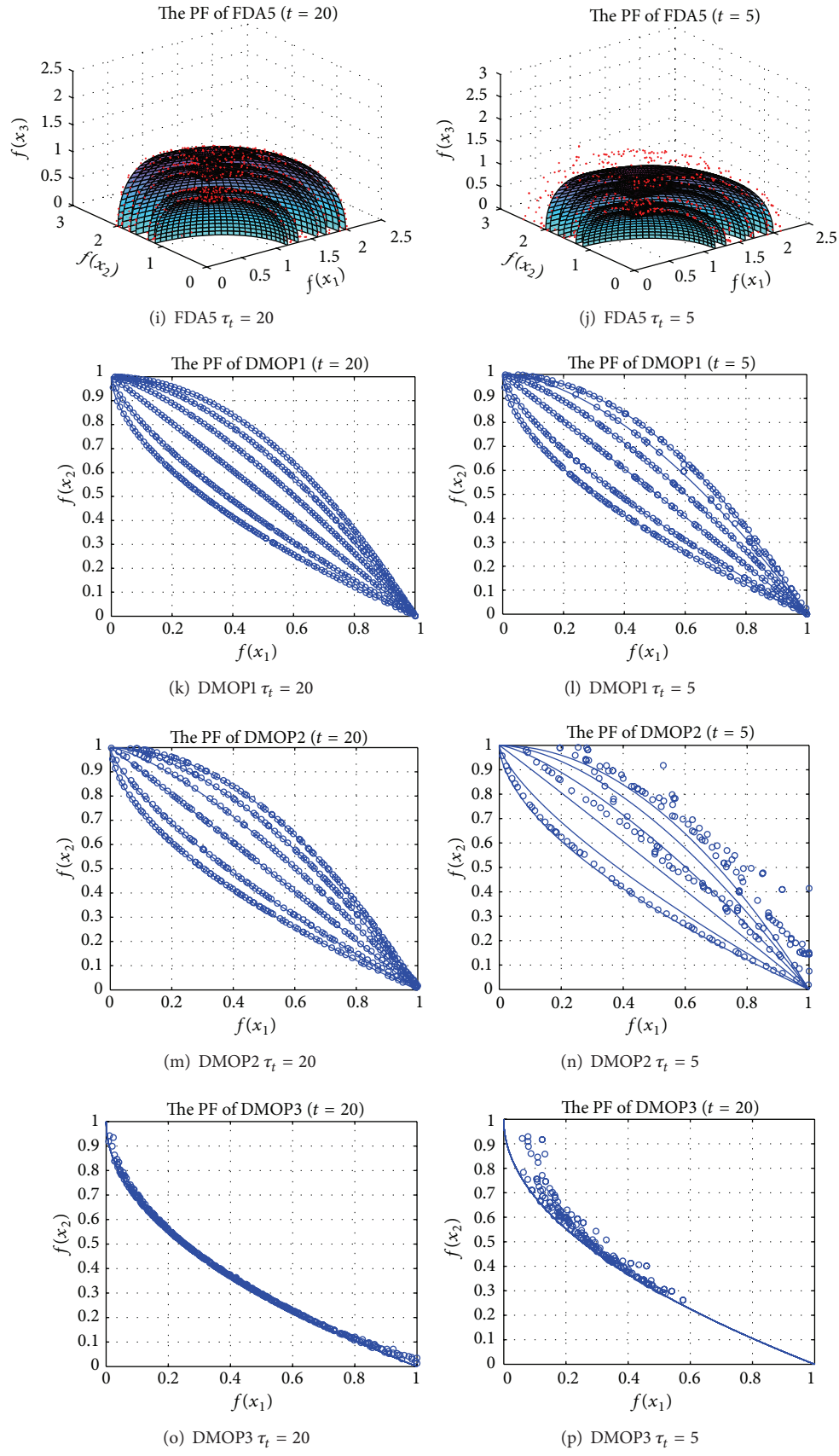
(a) FDA1 $\tau_t = 20$ (b) FDA1 $\tau_t = 5$

(c) FDA2 $\tau_t = 20$ (d) FDA2 $\tau_t = 5$

(e) FDA3 $\tau_t = 20$ (f) FDA3 $\tau_t = 5$

(g) FDA4 $\tau_t = 20$ (h) FDA4 $\tau_t = 5$

Figure 5: Continued.



(i) FDA5 $\tau_t = 20$ (j) FDA5 $\tau_t = 5$

(k) DMOP1 $\tau_t = 20$ (l) DMOP1 $\tau_t = 5$

(m) DMOP2 $\tau_t = 20$ (n) DMOP2 $\tau_t = 5$

(o) DMOP3 $\tau_t = 20$ (p) DMOP3 $\tau_t = 5$

Figure 5: The pareto fronts of benchmarks under different frequencies when $t = 5, 10, 20, 23, 26, 34$.

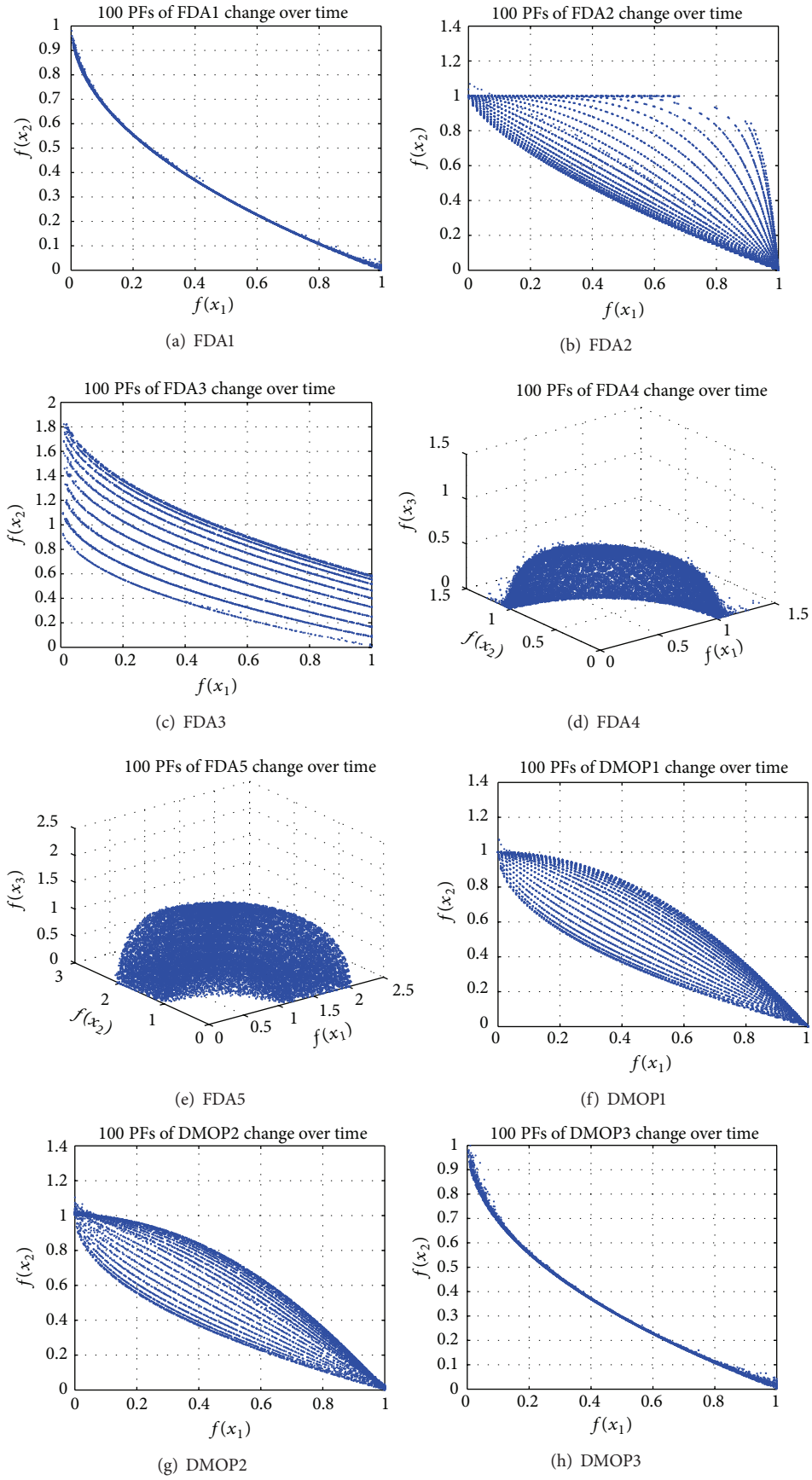
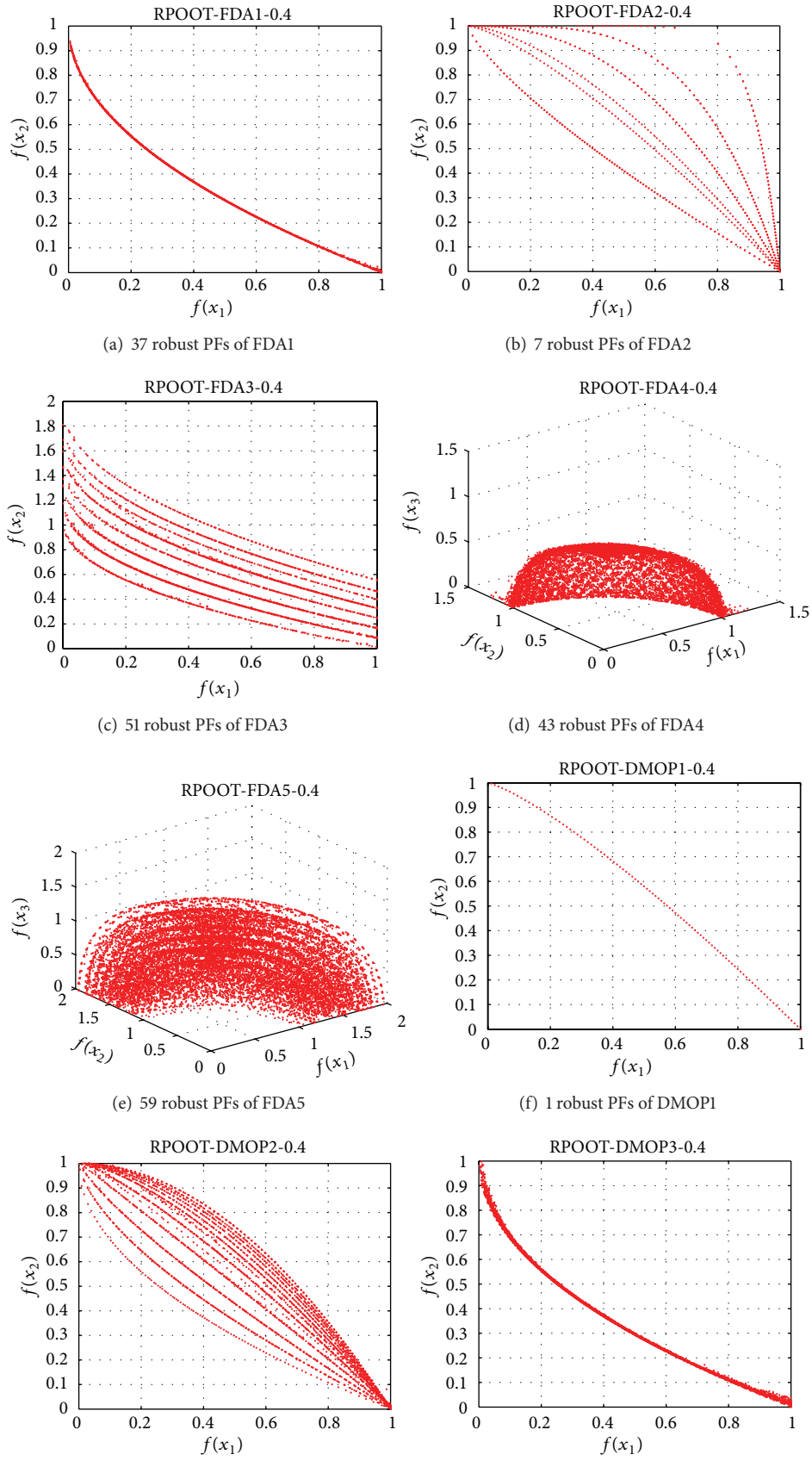


FIGURE 6: The 100 PFs of benchmark functions changing over time from $t = 1$ to 100.



(a) 37 robust PFs of FDA1

(b) 7 robust PFs of FDA2

(c) 51 robust PFs of FDA3

(d) 43 robust PFs of FDA4

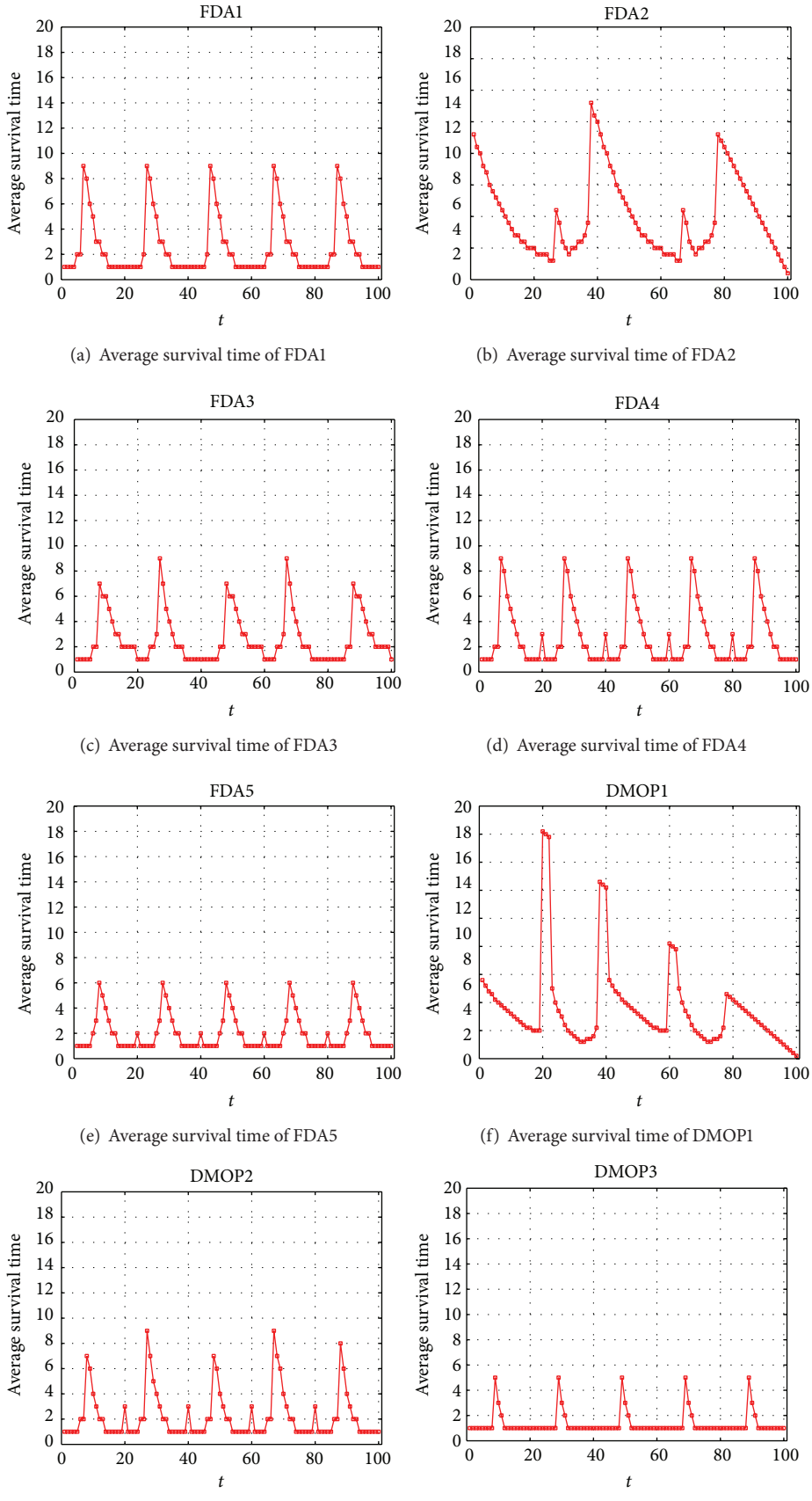
(e) 59 robust PFs of FDA5

(f) 1 robust PFs of DMOP1

(g) 40 robust PFs of DMOP2

(h) 80 robust PFs of DMOP3

Figure 7: The robust optimal pareto fronts of benchmark functions.



(a) Average survival time of FDA1

(b) Average survival time of FDA2

(c) Average survival time of FDA3

(d) Average survival time of FDA4

(e) Average survival time of FDA5

(f) Average survival time of DMOP1

(g) Average survival time of DMOP2

(h) Average survival time of DMOP3

Figure 8: Average survival time of benchmark functions.

Table 3: Comparison of the robust generation distances of robust optimal pareto fronts under different η.

Functions	Measures		η							
			$\eta = 0.1$	$\eta = 0.2$	$\eta = 0.3$	$\eta = 0.4$	$\eta = 0.5$	$\eta = 0.6$	$\eta = 0.7$	$\eta = 0.8$
FDA1	RGD	μ	0.0064	0.0067	0.0066	0.0054	0.0050	0.0046	0.0050	0.0043
		σ	0.0014	0.0010	0.0013	$6.2714e-4$	$4.1899e-4$	$7.0798e-4$	$2.2421e-4$	$1.2005e-4$
	RIGD	μ	0.1258	0.1028	0.1079	0.0095	0.0096	0.0100	0.0094	0.0077
		σ	0.1012	0.0996	0.0623	0.0021	0.0025	0.0050	0.0022	0.0016
FDA2	RGD	μ	0.0493	0.0802	0.2016	0.2044	0.2147	0.2359	0.2510	0.2397
		σ	0.0082	0.0055	0.0256	0.0186	$3.3963e-6$	0.0128	0.0148	0.0236
	RIGD	μ	0.0486	0.0822	0.2416	0.2438	0.2613	0.2675	0.2772	0.2762
		σ	0.0113	0.0049	0.0388	0.0316	$8.0722e-7$	0.0037	0.0148	0.0140
FDA3	RGD	μ	0.0438	0.0611	0.0689	0.1205	0.3034	0.3363	0.2115	0.3202
		σ	0.0183	0.0039	0.0032	0.0084	0.1128	0.1459	0.0240	0.0984
	RIGD	μ	0.1346	0.1342	0.1308	0.1544	0.3095	0.3452	0.2337	0.3341
		σ	0.0714	0.0608	0.0742	0.0579	0.1003	0.1544	0.0299	0.1021
FDA4	RGD	μ	0.3935	0.3898	0.3886	0.3879	0.3874	0.3877	0.3877	0.3869
		σ	0.0045	0.0030	$7.0914e-4$	$4.6005e-4$	0.0017	$8.8703e-4$	$4.4630e-4$	$5.2637e-4$
	RIGD	μ	0.0699	0.0573	0.0520	0.0543	0.0497	0.0476	0.0496	0.0424
		σ	0.0140	0.0131	0.0021	0.0043	0.0049	0.0033	0.0051	0.0023
FDA5	RGD	μ	0.7700	0.7703	0.7698	0.7761	0.7640	0.7650	0.7700	0.7637
		σ	0.0055	0.0061	0.0059	0.0172	0.0066	0.0060	0.0073	0.0074
	RIGD	μ	0.0897	0.0879	0.2593	0.3619	0.1659	0.1737	0.1984	0.1973
		σ	0.0063	0.0044	0.0506	0.1683	0.0020	0.0048	0.0019	0.0018
DMOP1	RGD	μ	0.0564	0.0913	0.1195	0.1523	0.1523	0.1523	0.1523	0.1523
		σ	0.0232	0.0080	0.0093	$2.2854e-6$	$3.8348e-6$	$3.1345e-6$	$3.8348e-6$	$3.4498e-6$
	RIGD	μ	0.0580	0.0845	0.1240	0.1564	0.1564	0.1564	0.1564	0.1564
		σ	0.0235	0.0083	0.0084	$1.0404e-7$	$1.3379e-7$	$1.2821e-7$	$1.3379e-7$	$1.7569e-7$
DMOP2	RGD	μ	0.0606	0.0611	0.0375	0.1255	0.0509	0.0634	0.0871	0.1198
		σ	0.0269	0.0193	0.0027	0.0671	0.0033	0.0134	0.0178	0.0594
	RIGD	μ	0.1779	0.1211	0.0789	0.1335	0.0518	0.0674	0.0866	0.1270
		σ	0.1300	0.0775	0.0553	0.0743	0.0030	0.0114	0.0218	0.0667
DMOP3	RGD	μ	0.0093	0.0098	0.0104	0.0101	0.0129	0.0107	0.0099	0.0107
		σ	$7.9206e-004$	0.0011	0.0022	0.0025	0.0063	0.0012	0.0013	0.0022
	RIGD	μ	0.1525	0.1847	0.1824	0.1465	0.1866	0.1517	0.2238	0.1993
		σ	0.0721	0.0407	0.0711	0.0508	$1.1782e-6$	0.0642	0.1010	0.1035

of the robust pareto-optimal over time is pointed out by the robustness definition of nondomination solution in the time scale. Secondly, we developed the new definition survival time which means how many time-varying environments it fits for. Thirdly, a framework for finding robust pareto fronts is proposed, and a MOEA/D is employed as an optimizer. Lastly, eight dynamic multiobjective benchmark functions are adopted to indicate the feasibility of the algorithm. From the simulation results, we receive that the robustness of RPOOT depends on the parameter η. Less robust pareto fronts are contained in the whole period with the increasing of the neighborhood size η. It means that the robustness is better. At the same time, the convergence of the RPOOT is worse. Moreover, the survival time of the robust pareto front at each time-varying moment is far larger than 1. That is, each robust pareto front of the robust pareto-optimal solution set can fit for more than 1 consecutive changed environments.

Due to the fact that the dynamic system is performed online, we do not obtain the future landscape on the current environment. The future fitness values of the solutions considered in RPOOT should be predicted through the past fitness values. So, the estimation and prediction task are inevitable in the future work.

Conflict of Interests

The authors declare that there is no conflict of interests regarding the publication of this paper.

Table 4: Comparison of the performance between TMO and RPOOT.

Functions		TMO		RPOOT ($\eta = 0.4$)	
		GD	IGD	RGD	RIGD
FDA1	μ	0.0016	0.0019	0.0054	0.0095
	σ	$6.3501e-007$	$5.9935e-008$	$6.2714e-4$	0.0021
FDA2	μ	0.0016	0.0019	0.2044	0.2438
	σ	$6.3501e-007$	$5.9935e-008$	0.0186	0.0316
FDA3	μ	0.0029	0.0030	0.1205	0.1544
	σ	$5.4816e-006$	$6.6723e-005$	0.0084	0.0579
FDA4	μ	0.0070	0.0067	0.3879	0.0543
	σ	$1.6289e-005$	$1.3751e-004$	$4.6005e-4$	0.0043
FDA5	μ	0.0119	0.0115	0.7761	0.3619
	σ	$1.1204e-005$	$1.3446e-004$	0.0172	0.1683
DMOP1	μ	0.0042	$2.6464e-005$	0.1523	0.1564
	σ	$1.2932e-006$	$1.3513e-006$	$2.2854e-6$	$1.0404e-7$
DMOP2	μ	$4.5087e-005$	$5.6999e-005$	0.1255	0.1335
	σ	$1.5932e-005$	$1.6302e-005$	0.0671	0.0743
DMOP3	μ	$3.8105e-005$	$6.7778e-005$	0.1763	0.1866
	σ	$6.4825e-006$	$8.7792e-006$	$1.4261e-5$	$1.4064e-6$

Table 5: The overall survival time of robust PFs.

Function	FDA1	FDA2	FDA3	FDA4	FDA5	DMOP1	DMOP2	DMOP3
Overall survival time	258	1009	262	273	199	1794	224	135

Acknowledgment

This research is jointly supported by the Fundamental Research Funds for the Central Universities (no. 2013XK09) and by National Basic Research Program of China (2014CB046300).

References

[1] M. Helbig and A. P. Engelbrecht, "Archive management for dynamic multi-objective optimisation problems using vector evaluated particle swarm optimisation," in *Proceedings of the IEEE Congress of Evolutionary Computation (CEC '11)*, pp. 2047–2054, New Orleans, La, USA, June 2011.

[2] C.-K. Goh and K. C. Tan, "A competitive-cooperative coevolutionary paradigm for dynamic multiobjective optimization," *IEEE Transactions on Evolutionary Computation*, vol. 13, no. 1, pp. 103–127, 2009.

[3] M. Farina, K. Deb, and P. Amato, "Dynamic multiobjective optimization problems: test cases, approximations, and applications," *IEEE Transactions on Evolutionary Computation*, vol. 8, no. 5, pp. 425–442, 2004.

[4] Y. Jin and B. Sendhof, "Constructing dynamic optimization test problems using the multi-objective optimization concept," in *Applications of Evolutionary Computing*, vol. 3005 of *Lecture Notes in Computer Science*, pp. 525–536, Springer, New York, NY, USA, 2004.

[5] T. T. Nguyen, S. Yang, and J. Branke, "Evolutionary dynamic optimization: a survey of the state of the art," *Swarm and Evolutionary Computation*, vol. 6, pp. 1–24, 2012.

[6] M. Cámara, J. Ortega, and F. de Toro, "A single front genetic algorithm for parallel multi-objective optimization in dynamic environments," *Neurocomputing*, vol. 72, no. 16–18, pp. 3570–3579, 2009.

[7] C. Hu, H. Yao, and X. Yan, "Multiple particle swarms coevolutionary algorithm for dynamic multi-objective optimization problems and its application," *Journal of Computer Research and Development*, vol. 50, no. 6, pp. 1313–1323, 2013.

[8] L. Wu, *Research on Multi-Objective Dynamic Differential Evolution and Its Applications*, Hunan University, Changsha, China, 2011.

[9] R. Shang, L. Jiao, Y. Ren, L. Li, and L. Wang, "Quantum immune clonal coevolutionary algorithm for dynamic multiobjective optimization," *Soft Computing*, vol. 18, no. 4, pp. 743–756, 2014.

[10] T. Ray, A. Isaacs, and W. Smith, "A memetic algorithm for dynamic multiobjective optimization," *Multi-Objective Memetic Algorithms*, vol. 171, pp. 353–367, 2009.

[11] H. Chen, M. Li, and X. Chen, "Using diversity as an additional-objective in dynamic multi-objective optimization algorithms," in *Proceedings of the 2nd International Symposium on Electronic Commerce and Security (ISECS '09)*, pp. 484–487, Nanchang, China, May 2009.

[12] A. Zhou, Y. Jin, and Q. Zhang, "A Population prediction strategy for evolutionary dynamic multiobjective optimization," *IEEE Transactions on Cybernetics*, vol. 44, no. 1, pp. 40–53, 2014.

[13] M. Li, S. Azarm, and V. Aute, "A multi-objective genetic algorithm for robust design optimization," in *Proceedings of Genetic and Evolutionary Computation Conference*, pp. 771–778, June 2005.

[14] K. Deb and H. Gupta, "Introducing robustness in multiple-objective optimization," KanGAL Report 2004016, Kanpur

Genetic Algorithms Laboratory, Indian Institute of Technology, Kanpur, India, 2004.

[15] C. Barrico and C. H. Antunes, "A new approach to robustness analysis in multi-objective optimization," in *Proceedings of the 7th International Conference on Multi-Objective Programming and Goal Programming*, Loire Valley (Tours), France, 2006.

[16] C. Barrico and C. H. Antunes, "Robustness analysis in multi-objective optimization using a degree of robustness concept," in *Proceedings of the IEEE Congress on Evolutionary Computation (CEC '06)*, pp. 1887–1892, Vancouver, Canada, July 2006.

[17] C. Cromvik, P. Lindroth, M. Patriksson, and A.-B. Strömberg, *A New Robustness Index for Multi-Objective Optimization Based on a User Perspective*, 2011.

[18] X. Yu, Y. Jin, K. Tang, and X. Yao, "Robust optimization over time—a new perspective on dynamic optimization problems," in *Proceedings of the IEEE Congress on Evolutionary Computation (CEC '10)*, pp. 3998–4003, July 2010.

[19] Y. Jin, K. Tang, X. Yu, B. Sendhoff, and X. Yao, "A framework for finding robust optimal solutions over time," *Memetic Computing*, vol. 5, no. 1, pp. 3–18, 2013.

[20] H. Fu, B. Sendhoff, K. Tang, and X. Yao, "Finding robust solutions to dynamic optimization problems," in *Applications of Evolutionary Computation*, vol. 7835 of *Lecture Notes in Computer Science*, pp. 616–625, Springer, Berlin, Germany, 2013.

[21] Q. Zhang and H. Li, "MOEA/D: a multiobjective evolutionary algorithm based on decomposition," *IEEE Transactions on Evolutionary Computation*, vol. 11, no. 6, pp. 712–731, 2007.

[22] C.-K. Goh and K.-C. Tan, *Evolutionary Multi-Objective Optimisation in Uncertain Environments: Issues and Algorithms*, Springer, Berlin, Germany, 2009.

Hindawi Publishing Corporation
Mathematical Problems in Engineering
Volume 2015, Article ID 837540, 15 pages
http://dx.doi.org/10.1155/2015/837540

Research Article

Tool-Body Assimilation Model Based on Body Babbling and Neurodynamical System

Kuniyuki Takahashi,[1] Tetsuya Ogata,[2] Hadi Tjandra,[1] Yuki Yamaguchi,[3] and Shigeki Sugano[1]

[1]*Graduate School of Creative Science and Engineering, Waseda University, Tokyo 1698555, Japan*
[2]*Graduate School of Fundamental Science and Engineering, Waseda University, Tokyo 1698555, Japan*
[3]*Graduate School of Informatics, Kyoto University, Kyoto 6068501, Japan*

Correspondence should be addressed to Kuniyuki Takahashi; takahashi@sugano.mech.waseda.ac.jp

Received 7 March 2014; Revised 15 June 2014; Accepted 15 June 2014

Academic Editor: Yi Chen

We propose the new method of tool use with a tool-body assimilation model based on body babbling and a neurodynamical system for robots to use tools. Almost all existing studies for robots to use tools require predetermined motions and tool features; the motion patterns are limited and the robots cannot use novel tools. Other studies fully search for all available parameters for novel tools, but this leads to massive amounts of calculations. To solve these problems, we took the following approach: we used a humanoid robot model to generate random motions based on human body babbling. These rich motion experiences were used to train recurrent and deep neural networks for modeling a body image. Tool features were self-organized in parametric bias, modulating the body image according to the tool in use. Finally, we designed a neural network for the robot to generate motion only from the target image. Experiments were conducted with multiple tools for manipulating a cylindrical target object. The results show that the tool-body assimilation model is capable of motion generation.

1. Introduction

Humans are capable of expanding their ability by using tools. Robots are expected to become more useful to society through the use of tools. With the development of robotics technology, robots have become very complex, with increasing numbers of applicable sensors and degrees of freedom. Therefore, complicated calculations are required to build conventional robot tool use models. Modeling robot tool use based on human cognitive development has been proposed as an approach to mitigate this problem [1]. Among the many factors of human cognitive development, tool-body assimilation, studied in the field of neuropsychology, has begun to gather attention [2]. Tool-body assimilation occurs when humans use a tool and treat it as an expansion of their own bodies. Iriki et al. recorded neurons called "bimodal neurons" before and after monkeys were trained to use a tool. Bimodal neurons respond both to tactile stimulation on the hand and to visual stimulation. Through tool use training, the visual receptive field of bimodal neurons expands from the monkey's hand to the surroundings of the grasping tool. This result shows that tool-body assimilation occurs at the neuron level. We aim to achieve robot tool use by this approach; by modeling tool-body assimilation, it is possible to alter the behavior of a posttrained body model by adding new neurons and expressing a "body model that is using a tool."

Tool use with tool-body assimilation is also gathering attention in the field of robotics. Nabeshima et al. [3] used visual and touch stimuli to connect bodily and sensory information. After training the relationships between visual and touch stimuli, dynamic touch was performed to predict the inertial parameters of the tool. The resulting simulation model allowed the robot to perform a pulling task with a target object located in an invisible area. However, the inertial parameters used as tool features were determined in advance, and the model was incapable of adapting to nonrigid bodies. Hikita et al. [4] treated tools as an expansion of the robot's

body in the form of saliency maps, basing their research on the experimental results of Iriki et al. However, no actual motion was generated.

There are various studies regarding tool manipulation, which is limited to not only focusing on tool-body assimilation, but also affordance. Affordance is the value that the environment provides to animals [5]. For example, a solid horizontal plane at knee height affords sitting. In the field of robotics, affordance research is divided into two approaches. One is the use of human affordance knowledge, while the second is the acquisition of affordance by the robot itself. Montesano et al. [6], Saxena et al. [7], and Song et al. [8] took the former approach, while Stoytchev [9], Detry et al. [10], and Tikhanoff et al. [11] adopted the latter approach. Montesano et al. trained a Bayesian network by giving the network information about the relations between actions and the resulting effects. Saxena et al. taught a robot where to grasp during tool use and generalized the model by categorizing unknown tools with similar shapes, allowing the grasping of unknown tools. Song et al. succeeded in training a model that grasps differently for different objectives. For example, this allows the robot to hold a cup from its side instead of grasping the top; open side of the cup when drinking is set as the objective. However, because it is very difficult for humans to directly teach a robot about the affordance of the environment, it is unrealistic to use this approach to teach about various numbers of tools. In addition, depending on the robot's bodily structure, it may be impossible to execute actions based on the trained affordance. In the study by Stoytchev, a robot learned the relationships between movements, tool shapes, and target objects by generating motions based on predetermined motion sets. Each tool was recognized by its respective colors, and the target object was moved to the target position by referring to the learned relationships. However, although the robot successfully learned the affordance of the tools, the generated motions were limited to the predetermined motions used during training. Detry et al. succeeded in achieving almost all possible graspable parts of various tools by testing almost all graspable positions. However, this is an unrealistic approach because large amounts of time will be required when this is done under real conditions, and the resulting model will not be able to adapt to unknown tools. Tikhanoff et al. performed object pulling tasks and taught pulling motions to the robot. In this case, the robot decided what tool and which part of the tool to use, based on the position of the target object. Because humans define the decision algorithm by using mathematical equations, appropriate modeling is required for new tasks.

The concerns described above are summarized as follows:

(1) Limited motion owing to predetermined motion patterns,

(2) Inability to adapt to new tools owing to predetermined tool and motion features,

(3) Large calculations required in generating target motions owing to the need to fully search all available parameters.

To overcome these concerns, this research will undertake the following approach with tool-body assimilation:

(1) body babbling with a humanoid model,

(2) recurrent neural network and deep neural networks,

(3) Motion generation by minimizing the error of the final state predicted from initial states.

Our purpose is to suggest the new method for tool use by robots by applying the concept of tool-body assimilation in cognitive science with body babbling and a neurodynamical system rather than modeling human's cognitive mechanism.

First we will describe the first step of the approach. Many past studies have built motion models in advance for motion generation and used teaching data that are designed by humans to train robots. This approach can adapt to conditions and tool functions that are known. However, the approach cannot adapt to unknown conditions. In addition, the creation of the teaching data is strongly influenced by the creator's knowledge and intentions, causing bias in the data. Because of this, it is difficult to effectively make use of the bodily constraints of the robot. It is hypothesized that the body has a significant influence on the development of intelligence [12]. During the early days of infants, the relationship between motion and the nervous system has yet to be modeled; thus, movements and interactions between the body and the environment cannot be reproduced. The infant learns the reproducible motions and the accompanying sensory information from this interaction. As the infant learns, passive motions are replaced with autonomous motions, and as the infant interacts with the environment, relations between primitive movements and sensations are learned. By learning these relationships, various information is gained, such as the postures of the body that are easy to move. Even in the case of tool use, knowledge related to the tool is earned and generalized through trial and error [13]. In the field of robotics, a phenomenon called body babbling, which is one of the steps in the infant development process, has recently began to gather attention [14–17]. Humans are capable of unlimited numbers of movement patterns due to having bodies with many degrees of freedom. However, the actual movement patterns that are used are limited owing to the constraints of the body structure, causing some degrees of freedom to be easier to move than others. In the case of tool use, there are unlimited ways to use tools. However, the actual use of a tool is limited to very few methods because tool use is also influenced by bodily constraints. Thus, it can be said that the way humans use tools has a strong relationship with the body [18]. When considering a model with human-like superior tool use capabilities, by using a body model that is similar to a human body, it is expected that the model will generate a human-like motion. In addition, considering human cognitive development, many trials of movements must be performed, and a body that is able to withstand these trials is needed. If this is done with real robots, the robot will break down. Therefore, the body babbling will be performed by a robot in a simulator.

We will now describe the second step of the approach. Past efforts in robotics have required the model of robot

movement, object manipulation, and features to be designed in advance. This method is highly effective in fixed environments such as factory production management. However, when considering robots that work in changing environments such as human living spaces, it is unrealistic to assume that the environment does not change. It is possible to partially overcome this problem by predicting the changes in the environment. However, the model would still not be able to adapt to unpredictable environmental changes. To overcome this issue, we took an approach in which the robot autonomously acquires features of data and learns the relationships between input and output from the sensory data that do not require preset features. With this approach, the model can adapt to new situations. Specifically, in this research, tool and motion features in the form of image features from camera data are self-organized by using deep neural networks (DNNs) with an autoencoder. The autoencoder can generate feature values automatically. This is done by training the autoencoder to give output values that are equal to the input values. The relations between the robot motion and the earned features are used to train a multiple time-scale recurrent neural network (MTRNN). With these two methods, it is possible to build a model that allows the robot to adapt to environmental changes and its own movements without predetermined information. In addition, as a characteristic of neural networks, these can estimate the output even when nontraining data are given based on the experiences that the neural networks have gained from training data.

We will now describe the third step of the approach. In the early days of infants, they tend to predict tool functions by using dynamic touch [5]. Dynamic touch refers to the movement of the body to acquire the characteristics of an object by moving the object [13]. Through tool manipulation experiences, humans tend to learn to use tool with tool-body assimilation. Nishide et al. [19] used a neural network model to build a tool-body assimilation model with dynamic touch. Because the tool features were self-organized with no settings determined in advance, the model is also applicable to untrained tools. The model predicted the features of the new tool, updated the parametric bias (PB) nodes connected to the neural network, and performed object pulling tasks. In their study, dynamic touch was required to differentiate between different tools. Michaels et al.'s experimental results imply that human estimates the tool function from the shape of tool as an overlay of tool use experiences [20]. If robots can also estimate tool function from the shape of tool, it should be possible to estimate tool function from the shape of unknown tools. Unlike Nishide et al.'s approach, our approach is to perform the estimation of the tool function from the image of the tool. Moreover, if the robot estimates the tool function from target state and generates motion with final states close to the target state, it will be useful because there is no necessity to teach the process of motion. In Nishide et al.'s study it is necessary to search motion parameters to generate target motion before motion generation. This results in large calculations. We further developed this model, allowing the model to predict tool functions from tool shapes, and designed the model to generate motion by minimizing errors between the final goal state and the final state predicted by the MTRNN.

The rest of this paper is composed as follows. In Section 2, the DNNs are discussed. In Section 3, an overview of the tool-body assimilation model is provided. In Section 4, the experimental setup is presented. In Section 5, the experimental results are given. In Section 6, we present our conclusions and describe future studies.

2. Deep Neural Networks

DNNs are multilayer neural networks, usually with five to 10 layers. DNNs allow highly precise image and speech recognition, so they have attracted attention in recent years [21, 22]. It is possible to reduce the dimensionality of data and to automatically extract features without predetermined information by using an autoencoder with a sand-glass-type DNN. It is possible to extract features from a hidden layer by substituting the unknown data in a trained DNN. DNNs are used in field of image recognition and speech recognition; however, they are not often used in the field of robotics [23, 24]. This is because for training DNNs require enormous, multidimensional training data. It is time consuming to acquire training data with robots, causing robots to break owing to use for a long time. To overcome these problems, we used a robot model in a simulator to perform random movements in the form of body babbling. This will be described in detail in Section 3, describing the tool-body assimilation model.

2.1. Training of Deep Neural Networks. A recent advance in learning for deep networks is to use layer-wise unsupervised pretraining methods [25]. Applying these methods before running fine-tuning methods overcome the difficulties related to deep learning. Martens [26] proposed the approach that developed a 2nd-order optimization method based on the Hessian free approach without using pretraining. This method is easy to use, effective, and efficient in training. Therefore, we adapted the learning method proposed by Martens.

2.2. Hessian-Free Optimization. Newton's method is the canonical 2nd-order optimization. The method is iteratively updates of the parameters $\theta \in \mathbb{R}^N$ of an objective function f by computing search vectors p and updating θ as $\theta + \alpha p$ for some α. The main idea of Newton's method is that f can be locally approximated around each θ, up to the 2nd-order, calculated as

$$f(\theta + p) \approx q_\theta(p) \equiv f(\theta) + \nabla f(\theta)^\top p + \frac{1}{2} p^\top B p, \quad (1)$$

where $B = H(\theta)$ is the Hessian matrix of f at θ. However, H can be indefinite so that (1) may not have a minimum. Moreover, if the values of p are large, the approximation of f becomes inaccurate. Therefore, H uses damping parameter λ for calculation to readjust $B = H + \lambda I$ for some constant $\lambda \geq 0$.

In standard Newton's method, $q_\theta(p)$ is optimized by computing the $N \times N$ matrix B and then solving the system

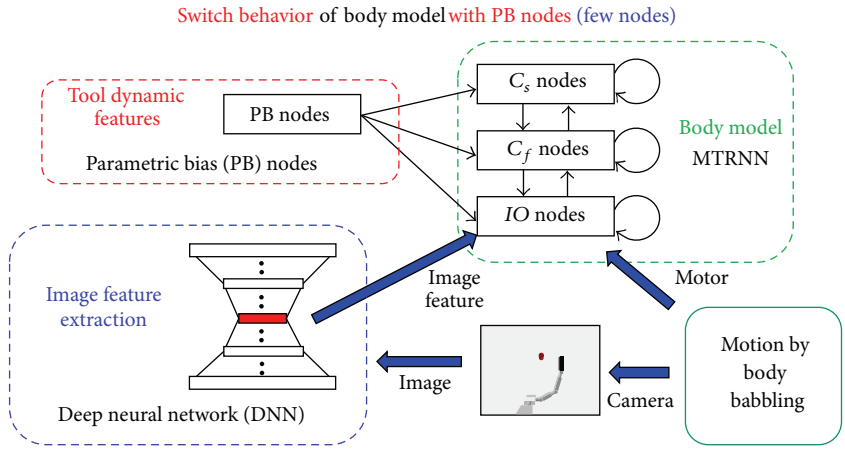


Figure 1: Overview of tool manipulation model.

$Bp = \nabla(\theta)^\top$. However, the calculation costs are very expensive when N is large even with modestly sized neural networks. Instead of this calculation, Martens develops the basis of the 2nd-order optimization approach with the linear conjugate gradient algorithm (CG) [26]. This optimizing quadratic objectives require only matrix vector products with B. Details about the implementation are provided in several other studies [26–28].

2.3. Competition with Self-Organizing Map. Some studies have used a self-organizing map (SOM) for the extraction of features [19, 29]. Arie et al. proved that SOM is the high compatibility with MTRNN. SOM is an unsupervised learning neural network proposed by Kohonen [30]. It is composed of input and output neurons. The neurons in the output layers are two-dimensionally arranged and possess weight vectors, w. The weight vectors are set to have the same dimensions (I) as the input vector, v. The image features are defined by the following formula:

$$p_i = \frac{\exp\left\{-\left\|w_i - v\right\|^2/\sigma\right\}}{\sum_{j\in N}\exp\left\{-\left\|w_j - v\right\|^2/\sigma\right\}}, \quad (2)$$

where N is the dimension of the SOM and $i \in I$.

It is possible to reduce the dimensionality of data by using an SOM. However, if there are many motion patterns for the robot, it is difficult to extract features using an SOM. Therefore, we used an autoencoder with the DNN for the extraction of features. This will be discussed in detail in Section 5, which describes the experimental results.

3. Tool-Body Assimilation Model

In this section, we provide an overview of the tool-body assimilation process. This process consists of three phases: (1) learning of the body model with random movement based on body babbling by a humanoid robot, (2) learning of tool dynamics features, and (3) generation of motion by using only a target image. Figure 1 shows an overview of the model. The model consists of three modules:

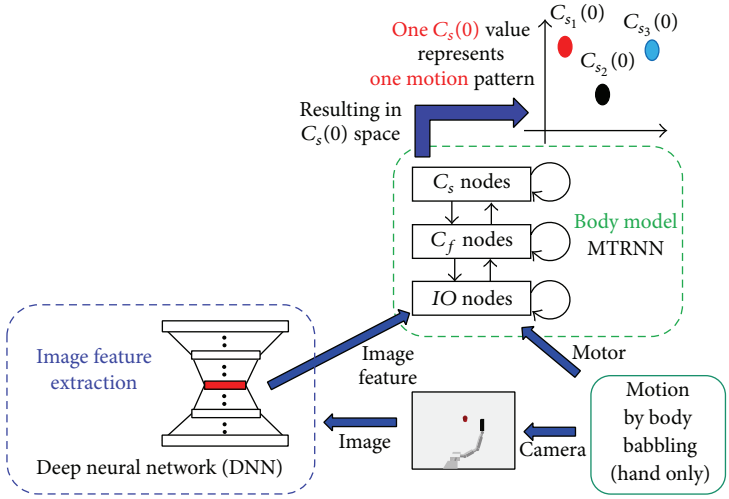
(i) body model module: MTRNN,

(ii) image feature extraction module: DNN,

(iii) tool dynamic feature module: PB-nodes.

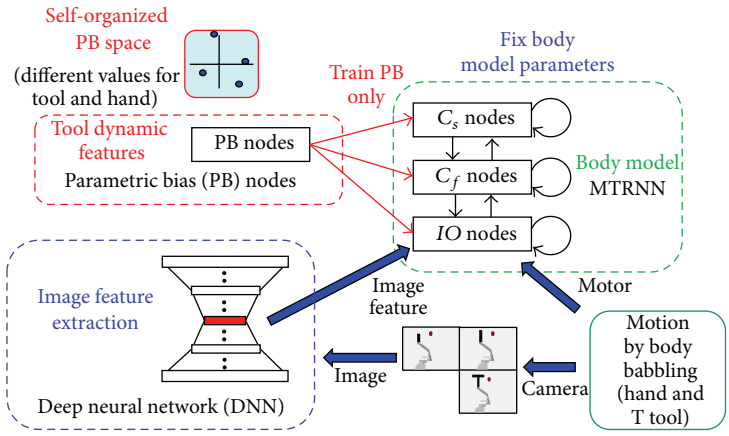
Figure 2 shows the learning process of tool-body assimilation. First, body babbling is performed with a bare hand, producing motor data and camera images. The image features are extracted with the DNN. Next, the MTRNN is trained by using these motor data and image features. Upon training, the MTRNN learns the body model of the robot. Next, by training only the PB nodes with different tool images, it is possible for the robot to learn the tool features without changing the body model. PB neurons are capable of learning how to modulate the body depending on the tool being used. This combination of MTRNN and PB nodes thus expresses "body model that uses tool." The number of PB nodes is much fewer than that needed for the body model (in this study, the number of PB nodes is five and body model requires 98 nodes), and the PB nodes represent the tool characteristics. Because the tool is treated as a part of the body, we expect the robot to use tools based on the experience of how to move the body. PB nodes do not represent the complicated methods of tool use itself but only show how to modulate the original body model. Therefore, even if the number of tools is increased, the number of neurons of PB nodes will still be fewer than in the body model. The time to learn the PB nodes is shorter than that needed for the body model. However, we suspect that the proposed model cannot learn all possible types of tools. This is mainly because of the difficulties in representing tools that cannot be treated as a modulation of the body. For example, tools with rotating motions such as a mixer will be a challenging tool to use.

3.1. Learning of Body Model Based on Body Babbling by a Humanoid Robot. In this phase, the robot, possessing a humanoid robot model, performs body babbling with a target object in its bare hand to learn its own body model. Figure 2(a) shows the learning process of body model. Firstly, the robot performs body babbling using bare hand, and motor data and camera images are produced. Then image

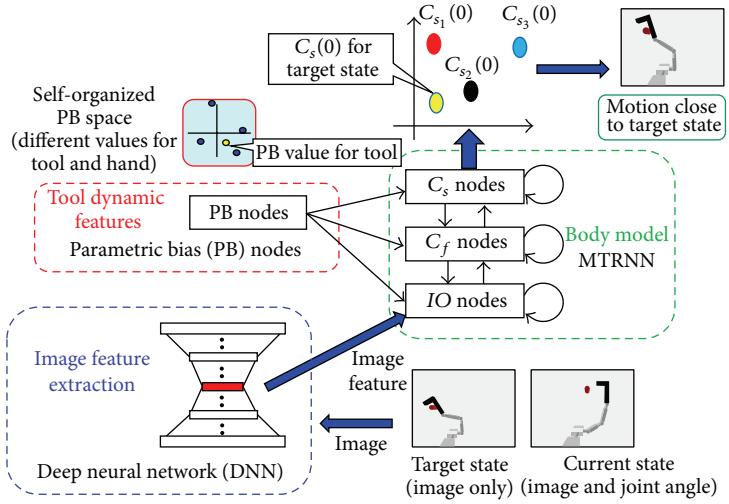


(a) Learning process of body model

PB nodes, which are of neuron number much smaller than the body model, represent the characteristic of tool



(b) Learning process of tool dynamics feature



(c) Process of motion generation

Figure 2: Learning process of tool-body assimilation.

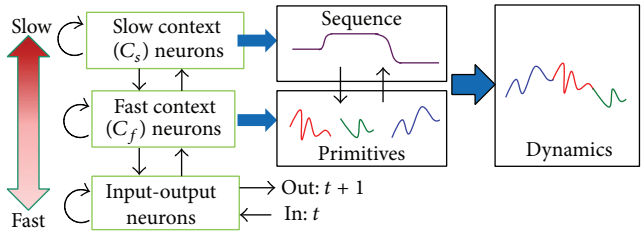


Figure 3: Dynamics representation of MTRNN.

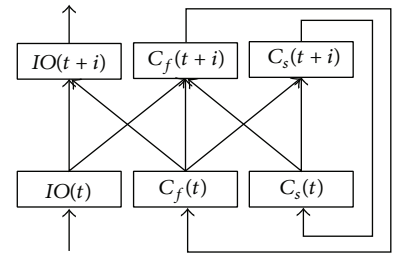


Figure 4: Composition of MTRNN.

features are extracted by the DNN. The relationship between motor and image features is learned by the MTRNN; that is, the robot learns its own body model.

Body babbling is a behavior observed in infants. Body babbling is considered as an explorative motion. Such motions that are related to human's intrinsic aspects such as motivation and preference are phenomena that are difficult to model. Hence, we modeled this as random motions in a similar way to the previous cognitive robotics studies [14–17]. By using the concept of body babbling, predetermined parameters are not required for performing motion by a robot. Body babbling requires numerous numbers of movements. Simulated experiments are effective, because it is difficult to perform many trials with real robots. During body babbling, sequential images and motor data are acquired. The features are extracted from the images by an image feature extraction function.

To adapt to unknown situations, the robot should have the ability to extract image features by itself. To achieve this, we propose the use of an autoencoder with the DNN extraction of image feature. We describe DNN in the next section. In this research, the input data to the DNN consists of the raw image pixels from the robot model's camera.

For the robot's body model, we implemented the MTRNN proposed by Yamashita and Tani [31]. The MTRNN is a kind of recurrent neural network (RNN) [32], which can predict the next state, IO($t + 1$), given the current state, IO(t). This MTRNN is capable of learning multiple sequential data. The MTRNN is composed of three types of neurons: fast context (C_f) nodes, slow context (C_s) nodes, and input-output (IO) nodes. Each type of node has a different time constant, representing the firing rate of the nodes. The faster (C_f) nodes learn the movement primitives of the data, whereas the slower (C_s) nodes learn the sequence of the data (Figure 3). The structure of the MTRNN proposed in this research is shown in Figure 4. During the learning of the MTRNN, the back propagation through time (BPTT) algorithm [33] is applied.

First, the output neurons are calculated by forward calculation. The internal value of the ith neuron at step t, $u_i(t)$, is calculated as

$$u_i(t) = \left(1 - \frac{1}{\tau_i}\right) u_i(t-1) + \frac{1}{\tau_i}\left[\sum_{j \in N} \omega_{ij} x_j(t-1)\right], \quad (3)$$

where τ_i is the time constant of the ith neuron, $x_j(t)$ is the input value of the ith neuron from the jth neuron, ω_{ij} is the weight value of the ith neuron from the jth neuron, and N is the number of neurons connecting to the ith neuron. The output of the $y_i(t)$ is calculated by multiplying the internal value and the sigmoid function:

$$y_i(t) = \text{sigmoid}(u_i(t)) = \frac{1}{1 + \exp(-u_i(t))}. \quad (4)$$

The input value is calculated as

$$x_i(t) = \begin{cases} \alpha \times y_i(t-1) + (1-\alpha) \times T_i(t) & i \in \text{IO} \\ y_i(t-1) & \text{otherwise,} \end{cases} \quad (5)$$

where $T_i(t)$ is the teacher signal and α is the feedback rate ($0 \le r \le 1$). The input value is calculated by adding the output value of the previous step to the teacher signal. This is done to avoid a diverging error during training. The inputs of the context layer are the outputs of the previous step.

The weight value is updated using the training error. The training error is calculated as

$$E = \sum_i \sum_{i \in \text{IO}} (y_i(t-1) - T_i(t))^2. \quad (6)$$

The weight from the ith neurons to the jth neurons is updated with the training error $\partial E / \partial \omega_{ij}$:

$$w_{ij}(n+1) = w_{ij} - \alpha \frac{\partial E}{\partial \omega_{ij}}, \quad (7)$$

where α is the training coefficient and n is the number of updates.

The initial value of C_s, $C_s(0)$, is also calculated by the BPTT algorithm:

$$Cs_i(n+1) = Cs_i(0) - \alpha \frac{\partial E}{\partial Cs_i(0)}. \quad (8)$$

After training, $Cs(0)$ represents the parameters of each learned sequence. Each sequence can be recovered by substituting the $Cs(0)$ values into the MTRNN.

By applying the BPTT algorithm to $C_s(0)$ with the fixed weights of MTRNN, it can be used as a recognition unit.

3.2. Learning of Tool Dynamics Features. In addition to the C_f, C_s, and IO nodes, PB nodes are connected to the MTRNN. The time constant of PB nodes is set to infinity; therefore, the values of the PB nodes do not change during

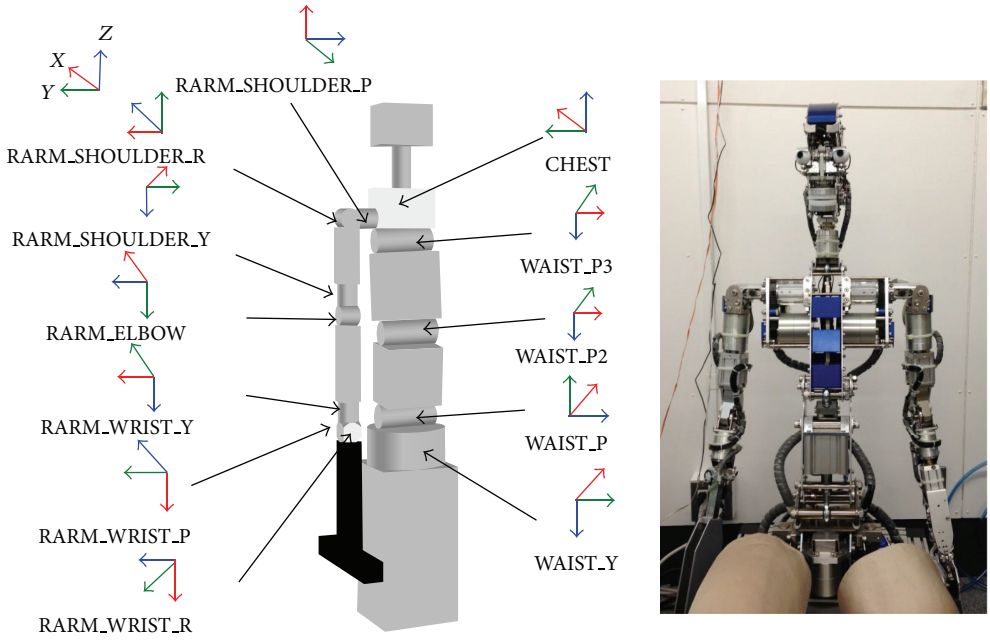


Figure 5: Robot based on ACTROID with T-shaped tool.

each sequence. During training, the weight of the MTRNN and $C_s(0)$ are fixed, and only the weights and values of the PB nodes are trained (Figure 2(b)). After training, the PB space that represents the tool dynamic features is formed. In other words, PB nodes are able to learn the visual changes resulting from differences in tool types. Then, the PB node applies bias to the body model according to the tool dynamic features that it learns, changing the body model's dynamics accordingly. This means that there is no need to retrain the robot's body model when a new tool is introduced. The value of the PB node is calculated by using the same method as for $C_s(0)$, and the weights from the PB nodes to other nodes are updated in the same manner as for other weights of the MTRNN.

3.3. Generation of Motion from Goal Image. In this phase, firstly, the initial image and joint angle are provided to the robot. With this the robot will be able to understand the environment's and the robot's own current state. Secondly, a target image is shown to the robot. During this time, the weights of the MTRNN and PB nodes are fixed. The $C_s(0)$ and PB values are calculated using the BPTT algorithm. Next, the differences between the target image and the associated image from the MTRNN are minimized. Using the PB and $C_s(0)$ values calculated with the algorithm, the MTRNN generates motor sequence data (Figure 2(c)). In Nishide et al.'s research, it was necessary to apply dynamic touch for tool type recognition. However, in this research, it is possible to recognize a tool from an image of the grasped tool.

4. Experimental Setup

4.1. Robot Model in Simulation. To evaluate the tool-body assimilation model, we built a humanoid robot model in the robotics simulator OpenHRP3 [34]. The model's size and

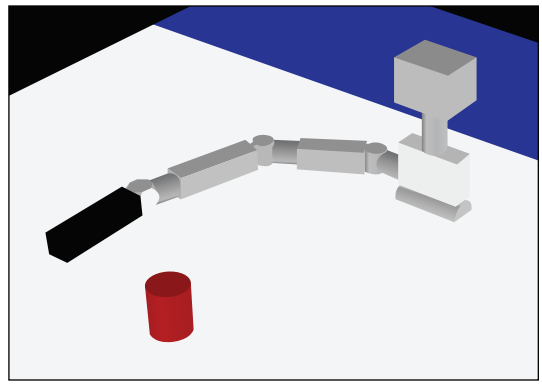


Figure 6: Experiment setting.

degrees of freedom (DOFs) were based on the humanoid robot ACTROID [35]. The range of motion of the model's joint angles was based on human [36]. To reduce calculation time, only the right arm, torso, and head of the robot were used. In addition, the left arm was removed and the legs were replaced with a box (Figure 5, Table 1).

4.2. Experimental Evaluation. In this experiment, an object pulling task with the robot's bare hand, an I-shaped tool, a T-shaped tool, an L-shaped tool, a J-shaped tool, "⊢" shaped tool, a C-shaped tool was used to evaluate the model (Figures 6 and 7). The size of the object was 0.08 [m] in diameter. The L-shaped tool was treated as an unknown tool, which is highly similar to the learned tools, and a J-shaped, "⊢" shaped, and C-shaped tool, which have high dissimilarities with the learned tools, are only used for evaluation and not for training. This task is commonly used in the study of robotic

Table 1: DOF and link length of the robot.

Link name (Arm)	a [mm]	α [deg.]	d [mm]	θ [deg.]	$q^{\max *}$ [deg.]	$q^{\min *}$ [deg.]
RARM_SHOULDER_P	0	90	−171	90	—	—
RARM_SHOULDER_R	0	90	0	90	120	−50
RARM_SHOULDER_Y	0	90	273	90	—	—
RARM_ELEBOW	−9	90	0	0	145	0
RARM_WRIST_Y	0	−90	240	90	—	—
RARM_WRIST_P	0	−90	0	−90	15	−55
RARM_WRIST_R	0	−90	0	0	—	—
(Tool)	Tool size	0	0	0	—	—

*Range of motion of the joint angle.

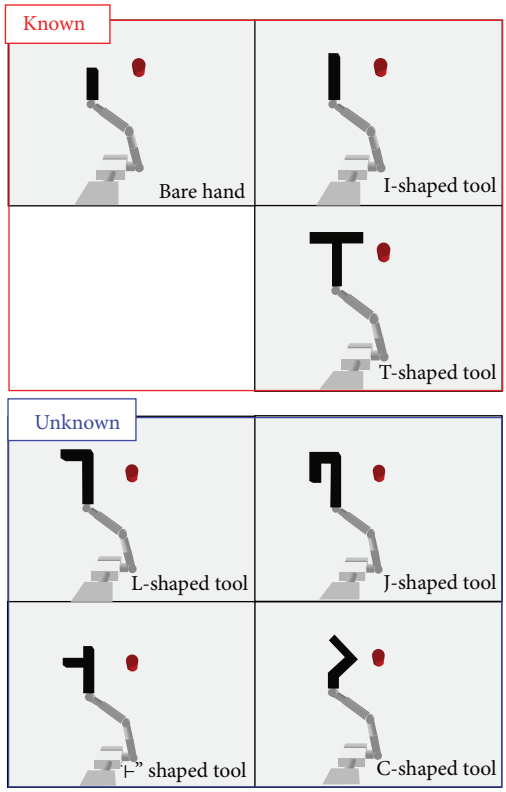


Figure 7: Tools used in experiment.

Figure 8: Hand posture used in motion pattern.

Table 2: Construction of DNN.

Dimensions of input-output nodes	768
Number of hidden layers	9
Dimensions of hidden nodes	500-250-100-50-15-50-100-250-500
Number of teaching data	13500

tool use and tool-body assimilation [3, 4, 9, 11, 19]. The robot performed body babbling in the presence of a target object (a cylinder) on a table for 6 [s] using its hand and a tool.

4.3. Motion by Body Babbling. To evaluate the effectiveness of this approach, the movement of the robot was confined to the plane of the desk (two-dimensional movements). In doing this, out of the seven DOFs of the robot's arm, only three DOFs were used. The robot's arm had two initial positions: to the left and to the right of the target object (Figure 8). For each initial position, the robot executed 75 sets of body babbling. The motions were generated by connecting the initial pose, the second pose, and the third pose. The second pose and third pose are selected randomly. The poses are connected by calculating the required acceleration with fifth-order linear interpolation. The accelerations of the beginning and end of the movement are calculated to be 0. Therefore, the movements become smooth and it is possible to control the motions according to the target motions. Although two initial positions are used for the teaching data in this study, parts of the motion paths in the training data are coded during the training of the RNN. This is synonymous to the learning of the varieties of trajectories. During body babbling, the robot obtained the teaching data that were used during training (Figure 9). The acquired data consisted of motor and image sequential data. The motor data of the three movable DOFs were recorded for 30 steps during the 6 [s] of random motion, that is, 7.5 [steps/s]. Image data constituted a gray-scale image of 32 × 24 pixels captured by a visual sensor on the robot. Twenty-five dimensions of the image features extracted from the image data by using an autoencoder with DNN, and three dimensions of the joint angles were used for the input data to train the MTRNN. The image data and joint angle were then normalized to [0.1, 0.9] and [0.0, 1.0] for use as the data for teaching the MTRNN. Table 2 shows the construction of the DNN. Table 3 shows the construction of the MTRNN.

Table 3: Construction of MTRNN.

Node name	Number of nodes	Time constant
Motor input nodes	3	2
Image feature input nodes	15	2
Fast context nodes	60	5
Slow context nodes	20	70
PB nodes	5	∞

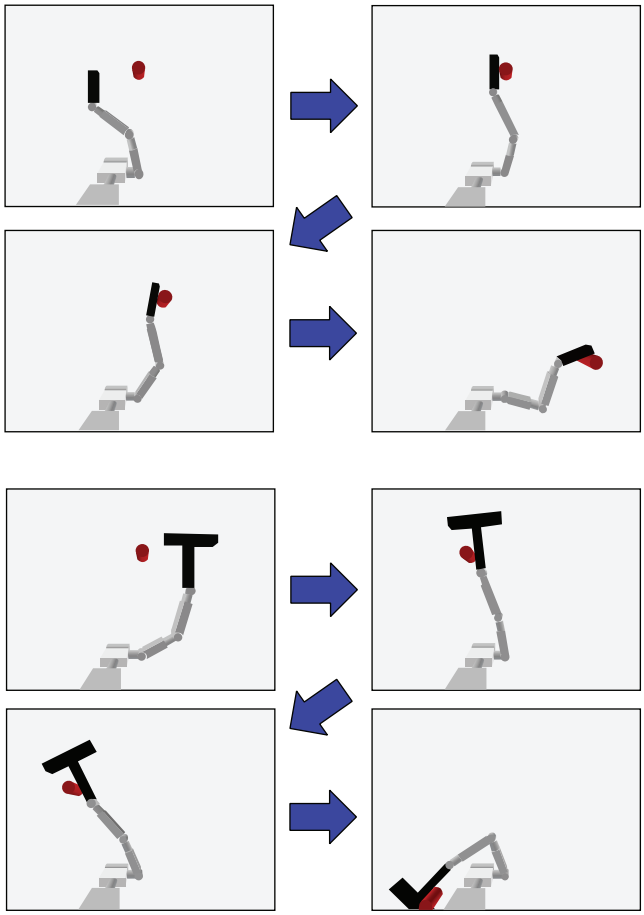


Figure 9: Body babbling with hand and T-shaped tool.

5. Experimental Results and Discussion

5.1. Extraction of Image Features by Self-Organizing Map. In previous studies, SOM has commonly been used as an image feature extraction method [19, 29, 37, 38]. To compare image features extraction by DNN, image features were also extracted by SOM. Figure 10 shows a visualization of the reference vectors of the SOM. Reference vectors are the visualization of image features. Reference vectors represent the patterns that are extracted from the input data. The characteristics of reference vectors are that the units that are mapped close to each other will have close resemblances to each other. In addition, each input data is classified to the locations of the reference vectors that are similar to the data. The dimensions of this SOM were 5 × 5. The results show

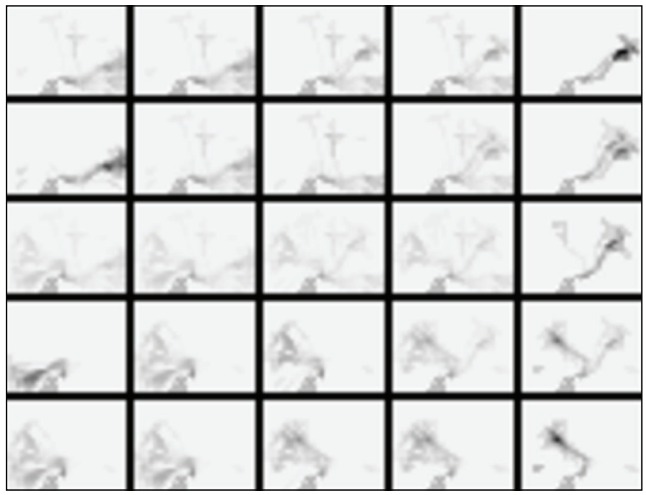


Figure 10: Reference vector of SOM.

Figure 11: Reference vector of SOM (bare hand and T-shaped tool only).

that the difference between the bare hand and tools is not learned accurately and that the motion patterns are not learned accurately. We changed the dimensions of SOM to 10 × 10. However, the motion patterns were not learned accurately after increasing the SOM dimension. Even if feature extraction is done well by increasing the dimension, it is difficult to learn by RNN because of the greater dimension. When more tools are introduced, the various tool conditions were included in each vector, causing the tool feature classification to fail. When only a few tools are used, it is possible to learn the image features accurately with SOM. This is shown in Figure 11, where image features of the bare hand and T-shaped tool were extracted by SOM.

5.2. Extraction of Image Features by DNN. Original images were recovered by substituting the image features extracted by DNN (Figure 12). The image of the bare hand and tools were accurately recovered. In addition, the position of the target object was recovered. Moreover, even with unknown

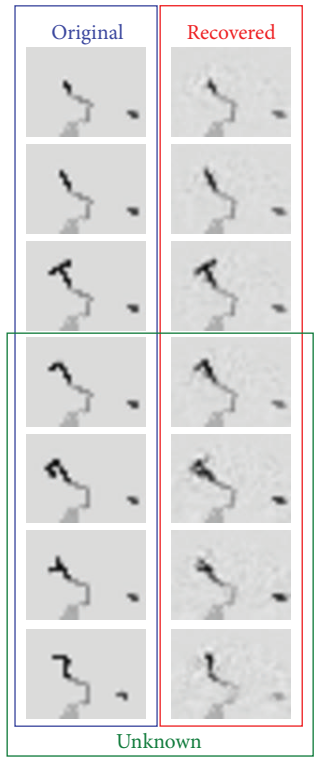


Figure 12: Original images recovered from DNN.

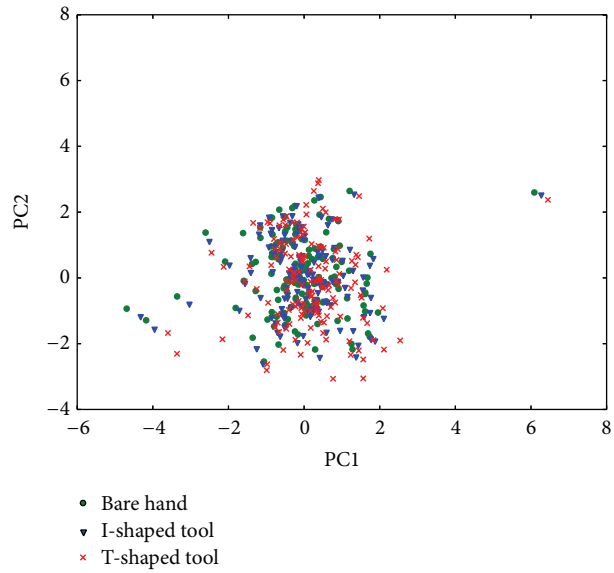


Figure 13: PCA PB space trained from the target image.

tools it was possible to recover the shapes of the unknown tools.

In the case of the SOM, extracted features of the target object were unclear; therefore, it was difficult to recover the position of the object accurately. DNN does not use classifications but instead makes use of autoencoders that are trained to produce the same output as the input data. With this, it is possible to reduce the dimensionality of large numbers of high-dimensional data followed by high reproduction performance. In addition, there is no need to fine tune the parameter settings with DNN. Thus, if there are large amounts of training data, the DNN is superior to the SOM in the extraction of image features.

5.3. Self-Organized Tool Function from PB Values. The principal component analysis (PCA) results of the PB values for the tool-body assimilation model are shown in Figure 13. The figure shows that self-organization failed for the features of each tool. Next, after body babbling was performed, as an additional condition we chose the motion patterns in which the bare hand and tools of the robot contacted the target object when moving between the two positions. PCA results of the PB values for the tool-body assimilation model are shown in Figure 14. The PB values are clustered based on the tool used during motion generation, that is, bare hand, I-shaped tool, and T-shaped tool. By these different conditions, we can see that the robot often moves without touching the target object when the additional conditions are not implemented. To generate motions there is no absolute need to gain tool functions. However, in this case the robot's adaptability to novel tools will decrease. To gain tool functions, we believe that it is important to actually have a sense of purpose to use the tool. The learning result implies that it is difficult to acquire the functions of the tool to just move the arm without purpose. We hypothesize that it is necessary to move with a purpose for acquiring tool functions. This idea is also described in a previous study [39]. When infants manipulate a grasped tool, it seems to be a random movement; however, we suspect that the intended movements are not performed correctly because their sensorimotor is not precisely formed. This problem will be solved by implementing explorative movements, and not random ones, for efficient learning.

Parts of the clusters are overlapping in Figure 14. In these overlapping regions, the robot manipulated the target object with part of its hand or a common part of tool (e.g., part of the "I" is common between the T-shaped and I-shaped tools) when the robot performed body babbling with a tool.

Figure 15 shows the clustering of the PB values after training of the model. As shown, each cluster is formed at a different area of the graph. Some parts of the clusters are overlapping. This is because when the robot generates the motion close to the target state by using parts of the tool that are similar to other tools, it distinguishes these as having the same tool function and chooses similar PB values. When the robot uses a different tool function, it chooses different PB values. PB values earned through recognition have large overlapping areas compared to the PB values earned during training. This is because the robot can generate the motion that uses the same tool function even if tool shape is different. It is considered that the robot generates motion with final states close to the target state which the robot has many experience with. The plots of PB values of unknown tools (L-shaped tool, J-shaped tool, "⊢" shaped tool, and C-shaped tool) overlap with each other and are hard to observe. This is because PB reflects not only the effect of tool shapes but also the tool functions used during each motion. For clarity's sake, we

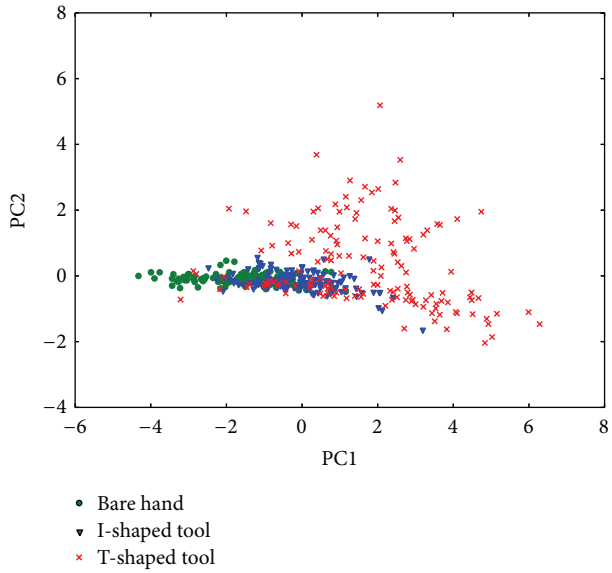


Figure 14: PCA PB space trained from the target image with additional conditions.

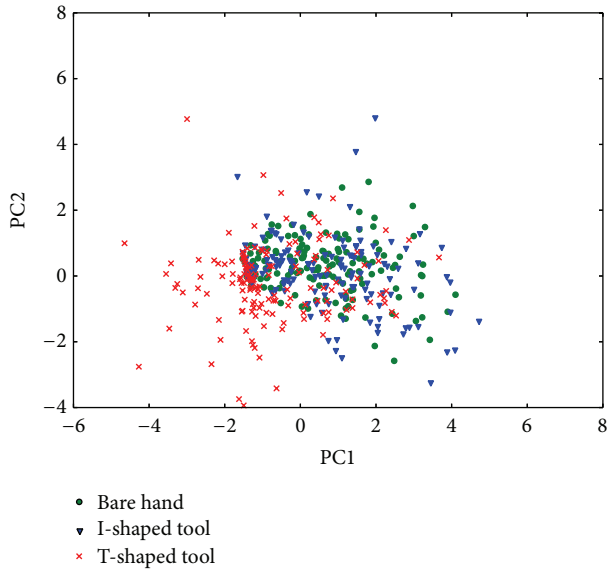


Figure 15: PCA PB space recognized (trained tools).

plotted the center of gravity of the PB of each tool (Figure 16). Variances of the PB values of the bare hand, I-, T-, L-, J-, "⊢", and C-shaped tool are 1.41, 3.17, 4.00, 3.87, 5.43, 6.89, and 3.32, respectively. Figure 16 shows that the PB values of the bare hand and I-shaped tools are similar. This is because the similar shapes of the bare hand and I-shaped tools lead to almost equivalent tool functions. With this observation, it can be said that similar shapes lead to similar usable tool functions during motions. It is observed that L-shaped tool has the intermediate tool functions of I-shaped and T-shaped tool. This is because L-shaped tool is similar to I-shaped and T-shaped tools. In the case of the J-shaped and C-shaped tools, the PC1 values are similar to the PC1 values of the

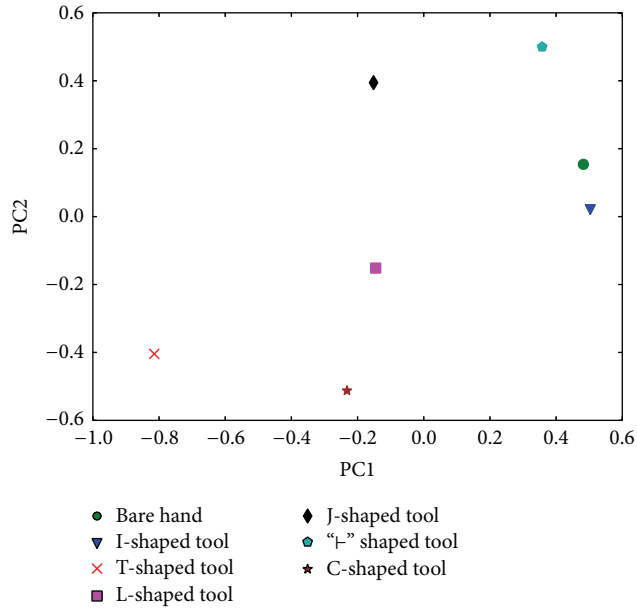


Figure 16: PCA PB space recognized (all tools).

L-shaped tool. Here, the difference of PC2 values of the L-shaped and C-shaped tools is smaller than the difference of PC2 values of the L-shaped and J-shaped. Thus, it can be said that the function of C-shaped tool is more similar to L-shaped tool than J-shaped tool. It is observed that the PB values of the "⊢" shaped tool are differ from other tools except the PC1 values of bare hand and I-shaped tool. Thus, it can be said that "⊢" shaped tool has part of the functions of bare hand and I-shaped tool and different functions when comparing other tools. With these observations, it can be said that the robot recognizes that the tools each have different tool functions.

5.4. Generated Motion. We evaluated the performance of our system by counting the number of the tasks which successfully moved the object to a position within the radius of R pixels from the goal position in the visible area (32×24 pixels). Figure 17 shows the relationships between the success rate and R. The success rate within $R = 2$ was about 20 to 35 percent; however within $R = 5$ it was more than 50 percent even if the robot used the untrained L-shaped tool, J-shaped tool, "⊢" shaped tool, and C-shaped tool. As one of the ways to improve the success rate, it may be better to learn gradually from coarse to fine images as shown by the experimental results of Kawai et al. [40].

Examples of motion generated by the robot when given a target image are shown in Figures 18, 19, 20, and 21. As shown, motions close to the target state were generated even if the robot uses an untrained tool. In Figure 17, it can be observed that it is possible to manipulate the object with the same accuracy as the learned tools even when using unknown tools. In addition, Figure 22 shows an example of motion that starts from an untrained initial pose and object position. As shown, motions close to the target state were generated even if the robot starts from an untrained initial state. As a matter of interest, in Figure 19 the robot generated the pulling

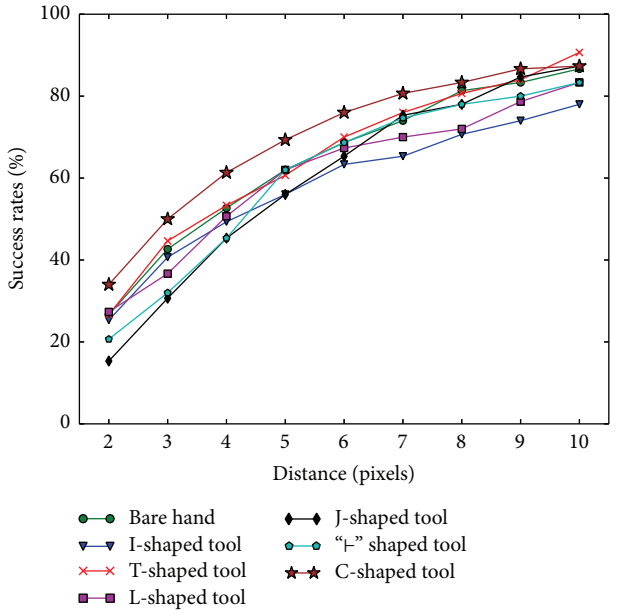


Figure 17: Relationships between the success rate and the position error.

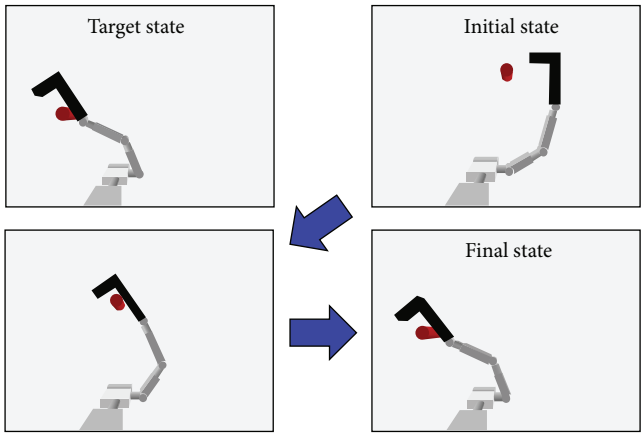


Figure 18: Generated motion by L-shaped tool from right side to object.

motion with a J-shaped tool after first avoiding contact of the protruding part of the tool and the object. If the robot did not avoid the contact at first, the object will be repelled and the robot would not be able to manipulate the object correctly.

Among the generated motions that have a final state close to a given target state, some have different movement courses compared to the teaching data (Figure 23). Because motions other than the learned ones are generated, it can be said that the model does not overfit. Figure 14 shows that the PB values formed different clusters for each tool. Figure 16 shows that robot recognizes different PB values for each tool. In such cases, it can be said that the robot have acquired the tool function (affordance) and generated the motion by using the tool function.

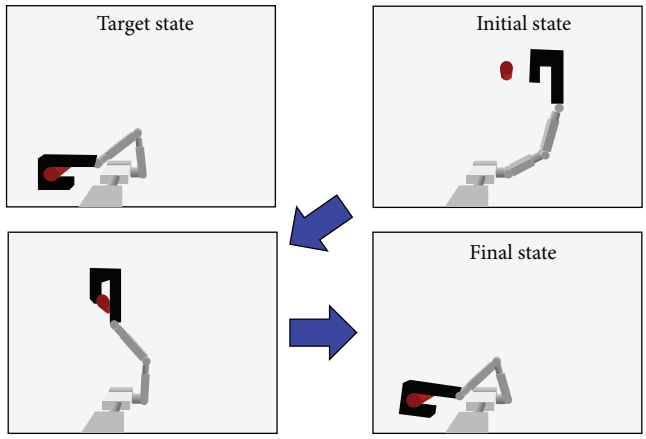


Figure 19: Generated motion by J-shaped tool from right side to object.

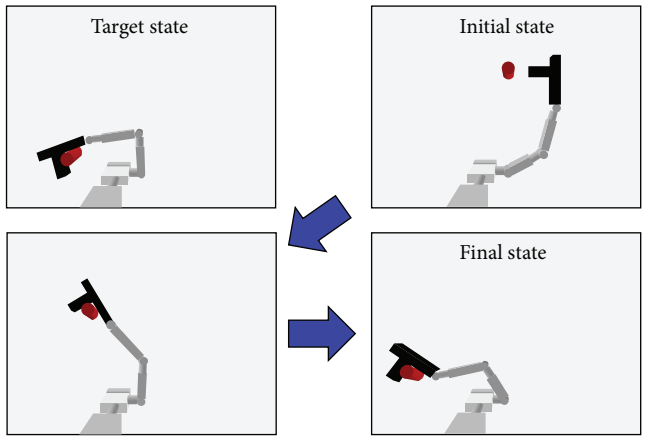


Figure 20: Generated motion by "⊢" shaped tool from right side to object.

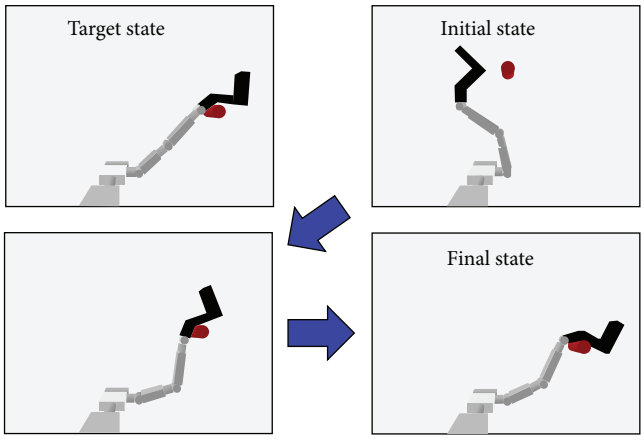


Figure 21: Generated motion by C-shaped tool from left side to object.

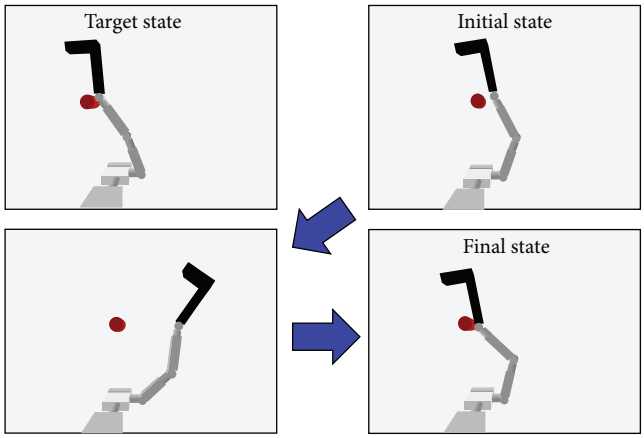


Figure 22: Generated motion by L-shaped tool from random initial state.

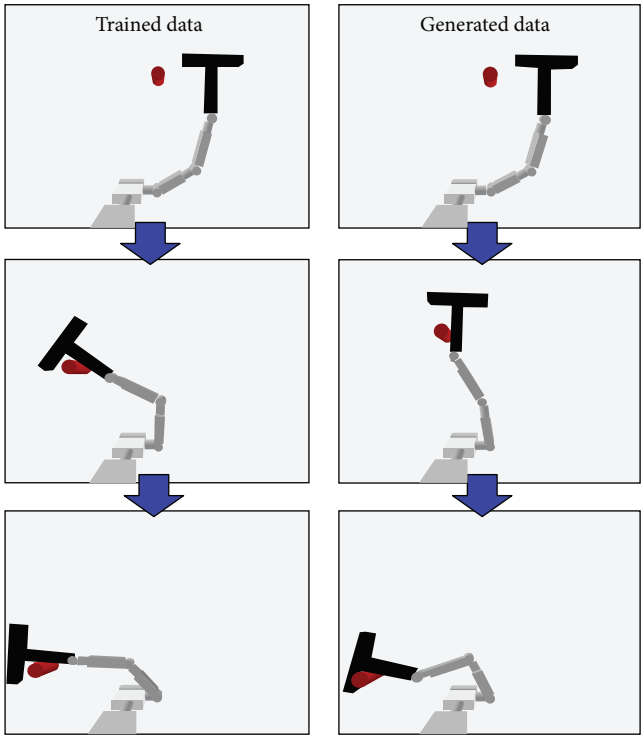


Figure 23: Generated motion by T-shaped tool on the way does not match.

6. Conclusion

This study's objective is to achieve robot tool use without the need for predetermined features and models by having the robot self-organize the required features inspired by human development and cognitive mechanisms. By using the concept of tool-body assimilation, it is possible to treat a tool as an extension of the body. Therefore, it is possible to represent the body and tool use models in one single model. Previous models implemented additional models per additional tool to be used in addition to the body model. However, our proposed model made it possible to represent all this with only one model. Related studies have required preset motions, preset tool and motion features, and full searches of all possible motions during motion generation. To overcome these issues, we propose the following approach: (1) body babbling with a humanoid model that does not require preset motions, (2) learning algorithm that does not require preset sensorimotor integration and tool features, with the concept of tool-body assimilation by using MTRNN and image feature extraction by an autoencoder with DNN, and (3) recognition of motion from the goal state. The evaluation experiment is an object manipulation task conducted with OpenHRP3, a robotics simulator. As a result, when given an image of a final state, the robot is able to generate a motion similar to the final state.

As next steps, we plan to extend the study to a seven-degrees-of-freedom model, design research settings that consider more of the human body, and set a more specific task for quantitative assessments.

Conflict of Interests

The authors declare that there is no conflict of interests regarding the publication of this paper.

Acknowledgments

The work has been supported by JST PRESTO "Information Environment and Humans," MEXT Grant-in-Aid for Scientific Research on Innovative Areas "Constructive Developmental Science" (24119003), JSPS Grant-in-Aid for Scientific Research (S)(2522005), "Fundamental Study for Intelligent Machine to Coexist with Nature" Research Institute for Science and Engineering, Waseda University, and Grants for Excellent Graduate Schools, MEXT, Japan. The authors would like to thank H. Arie, K. Noda, and S. Murata for their help in conducting the experiments.

References

[1] M. Asada, K. Hosoda, Y. Kuniyoshi et al., "Cognitive developmental robotics: a survey," *IEEE Transactions on Autonomous Mental Development*, vol. 1, no. 1, pp. 12–34, 2009.

[2] A. Maravita and A. Iriki, "Tools for the body (schema)," *Trends in Cognitive Sciences*, vol. 8, no. 2, pp. 79–86, 2004.

[3] C. Nabeshima, Y. Kuniyoshi, and M. Lungarella, "Towards a model for tool-body assimilation and adaptive tool-use," in *Proceedings of the IEEE 6th International Conference on Development and Learning (ICDL '07)*, pp. 288–293, July 2007.

[4] M. Hikita, S. Fuke, M. Ogino, and M. Asada, "Cross-modal body representation based on visual attention by saliency," in *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS '08)*, pp. 2041–2046, September 2008.

[5] J. J. Gibson, *The Senses Considered as Perceptual Systems*, Houghton-Mifflin Company, Boston, Mass, USA, 1966.

[6] L. Montesano, M. Lopes, A. Bernardino, and J. Santos-Victor, "Learning object affordances: from sensory—motor coordination to imitation," *IEEE Transactions on Robotics*, vol. 24, no. 1, pp. 15–26, 2008.

[7] A. Saxena, J. Driemeyer, and A. Y. Ng, "Robotic grasping of novel objects using vision," *International Journal of Robotics Research*, vol. 27, no. 2, pp. 157–173, 2008.

[8] D. Song, K. Huebner, V. Kyrki, and D. Kragic, "Learning task constraints for robot grasping using graphical models," in *Proceeding of the 23rd IEEE/RSJ 2010 International Conference on Intelligent Robots and Systems (IROS '10)*, pp. 1579–1585, Taipei, Taiwan, October 2010.

[9] A. Stoytchev, "Behavior-grounded representation of tool affordances," in *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA '05)*, pp. 3060–3065, April 2005.

[10] R. Detry, E. Başeski, M. Popović et al., "Learning object-specific grasp affordance densities," in *Proceedings of the IEEE 8th International Conference on Development and Learning (ICDL '09)*, pp. 1–7, June 2009.

[11] V. Tikhanoff, U. Pattacini, L. Natale, and G. Metta, "Exploring affordances and tool use on the iCub," in *Proceedings of the IEEE/RAS International Conference of Humanoids Robotics*, 2013.

[12] R. Pfeifer and J. Bongard, *How the Body Shapes the Way We Think: A New View of Intelligence*, MIT Press, Cambridge, Mass, USA, 2007.

[13] A. Streri and J. Féron, "The development of haptic abilities in very young infants: from perception to cognition," *Infant Behavior and Development*, vol. 28, no. 3, pp. 290–304, 2005.

[14] R. Saegusa, G. Metta, G. Sandini, and S. Sakka, "Active motor babbling for sensorimotor learning," in *Proceedings of the IEEE International Conference on Robotics and Biomimetics (ROBIO '08)*, pp. 794–799, 2009.

[15] K. Mochizuki, S. Nishide, H. G. Okuno, and T. Ogata, "Developmental human-robot imitation learning of drawing with a neuro dynamical system," in *Proceedings of the International Conference on Systems, Man, and Cybernetics*, pp. 2337–2341, 2013.

[16] D. Caligiore, T. Ferrauto, D. Parisi, N. Accornero, M. Capozza, and G. Baldassarre, "Using motor babbling and hebb rules for modeling the development of reaching with obstacles and grasping," in *Proceedings of the International Conference on Cognitive Systems (COGSYS '08)*, 2008.

[17] J. Sturm, C. Plagemann, and W. Burgard, "Unsupervised body scheme learning through self-perception," in *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA '08)*, pp. 3328–3333, Pasadena, Calif, USA, May 2008.

[18] T. Sasaoka, H. Mizuhara, and T. Inui, "The interaction between the parietal and motor areas in dynamic imagery manipulation: an fMRI study ," in *Advances in Cognitive Neurodynamics (II)*, pp. 345–349, 2011.

[19] S. Nishide, J. Tani, T. Takahashi, H. G. Okuno, and T. Ogata, "Tool-body assimilation of humanoid robot using a neurodynamical system," *IEEE Transactions on Autonomous Mental Development*, vol. 4, no. 2, pp. 139–149, 2012.

[20] C. F. Michaels, Z. Weier, and S. J. Harrison, "Using vision and dynamic touch to perceive the affordances of tools," *Perception*, vol. 36, no. 5, pp. 750–772, 2007.

[21] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "ImageNet classification with deep convolutional neural networks," *NIPS Proceedings*, vol. 1, no. 2, 2012.

[22] G. Hinton, L. Deng, D. Yu et al., "Deep neural networks for acoustic modeling in speech recognition: the shared views of four research groups," *IEEE Signal Processing Magazine*, vol. 29, no. 6, pp. 82–97, 2012.

[23] K. Noda, H. Arie, Y. Suga, and T. Ogata, "Intersensory causality modeling using deep neural networks," in *Proceedings of the IEEE International Conference on Systems, Man, and Cybernetics (SMC '13)*, Manchester, UK, October 2013.

[24] K. Noda, H. Arie, Y. Suga, and T. Ogata, "Multimodal integration learning of object manipulation behaviors using deep neural networks," in *Proceedings of IEEE-RSJ International Conference on Intelligent Robots and Systems (IROS '13)*, IEEE/RSJ, Tokyo, Japan, 2013.

[25] G. E. Hinton and R. R. Salakhutdinov, "Reducing the dimensionality of data with neural networks," *Science*, vol. 313, no. 5786, pp. 504–507, 2006.

[26] J. Martens, "Deep learning via Hessian-free optimization," in *Proceedings of the 27th International Conference on Machine Learning (ICML '10)*, pp. 735–742, June 2010.

[27] B. Pearlmutter, "Fast exact multiplication by the Hessian," *Neural Computation*, vol. 6, no. 1, pp. 147–160, 1994.

[28] N. N. Schraudolph, "Fast curvature matrix-vector products for second-order gradient descent," *Neural Computation*, vol. 14, no. 7, pp. 1723–1738, 2002.

[29] H. Arie, T. Endo, T. Arakaki, S. Sugano, and J. Tani, "Creating novelgoal-directed actions at criticality: a neuro-robotic experiment," *New Mathematics and Natural Computation*, vol. 5, pp. 307–334, 2009.

[30] T. Kohonen, *Self-Organization and Associative Memory*, vol. 8, Springer, New York, NY, USA, 2nd edition, 1988.

[31] Y. Yamashita and J. Tani, "Emergence of functional hierarchy in a multiple timescale neural network model: a humanoid robot experiment," *PLoS Computational Biology*, vol. 4, no. 11, Article ID e1000220, 2008.

[32] M. Jordan, "Attractor dynamics and parallelism in a connectionist sequential machine," in *Proceedings of the 8th Annual Conference of the Cognitive Science Society*, pp. 513–546, Erlbaum, Hillsdale, NJ, USA, 1986.

[33] D. Rumelhart, G. Hinton, and R. Williams, "Learning internal representation by error propagation," in *Parallel Distributed Processing*, D. E. Rumelhart and J. L. McLelland, Eds., MIT Press, Cambridge, Mass, USA, 1986.

[34] F. Kanehiro, H. Hirukawa, and S. Kajita, "OpenHRP: open architecture humanoid robotics platform," *International Journal of Robotics Research*, vol. 23, no. 2, pp. 155–165, 2004.

[35] "Kokoro: custom-made robot: ACTROID," 2014, http://www.kokoro-dreams.co.jp/rt_tokutyu/actroid.html.

[36] I. A. Kapandji, "Physiologie Articulaire," MALOINE S.A. EDITEUR, 1980.

[37] K. Takahashi, T. Ogata, H. Tjandra, Y. Yamaguchi, Y. Suga, and S. Sugano, "Tool—body assimilation model using a neurodynamical system for acquiring representation of tool function and motion," in *IEEE/ASME International Conference on Advanced Intelligent Mechatronics (AIM '14)*, Besancon, France, July 2014.

[38] K. Takahashi, T. Ogata, H. Tjandra, S. Murata, H. Arie, and S. Sugano, "Tool-body assimilation model based on body babbling and a neuro-dynamical system for motion generation," in

Proceedings of the 24th International Conference on Artificial Neural Networks (ICANN '14), Lecture Notes in Computer Science, Hamburg, Germany, September 2014.

[39] J. Namikawa, R. Nishimoto, and J. Tani, "A neurodynamic account of spontaneous behaviour," *PLoS Computational Biology*, vol. 7, no. 10, Article ID e1002221, 2011.

[40] Y. Kawai, Y. Nagai, and M. Asada, "Perceptual development triggered by its self-organization in cognitive learning," in *Proceeding of the 25th IEEE/RSJ International Conference on Robotics and Intelligent Systems (IROS '12)*, pp. 5159–5164, Vilamoura, Portugal, October 2012.

Hindawi Publishing Corporation
Mathematical Problems in Engineering
Volume 2014, Article ID 452803, 14 pages
http://dx.doi.org/10.1155/2014/452803

Research Article

Feature Based Stereo Matching Using Two-Step Expansion

Liqiang Wang,[1] Zhen Liu,[1] and Zhonghua Zhang[2]

[1] *School of Instrumentation Science & Opto-Electronics Engineering, Beihang University, Beijing 100191, China*
[2] *National Institute of Metrology, Beijing 100029, China*

Correspondence should be addressed to Liqiang Wang; gueij@163.com

Received 2 January 2014; Revised 19 June 2014; Accepted 21 July 2014; Published 18 December 2014

Academic Editor: Yi Chen

This paper proposes a novel method for stereo matching which is based on image features to produce a dense disparity map through two different expansion phases. It can find denser point correspondences than those of the existing seed-growing algorithms, and it has a good performance in short and wide baseline situations. This method supposes that all pixel coordinates in each image segment corresponding to a 3D surface separately satisfy projective geometry of 1D in horizontal axis. Firstly, a state-of-the-art method of feature matching is used to obtain sparse support points and an image segmentation-based prior is employed to assist the first region outspread. Secondly, the first-step expansion is to find more feature correspondences in the uniform region via initial support points, which is based on the invariant cross ratio in 1D projective transformation. In order to find enough point correspondences, we use a regular seed-growing algorithm as the second-step expansion and produce a quasi-dense disparity map. Finally, two different methods are used to obtain dense disparity map from quasi-dense pixel correspondences. Experimental results show the effectiveness of our method.

1. Introduction

Stereo matching is an international research focus of computer vision [1]. It can produce a disparity map from stereo images which are captured by cameras in different viewpoints. This technology is important in 3D reconstruction, virtual view rendering, and automatic navigation. It is a key point to know how to compute a precise disparity map in a complex environment by stereo matching. There is much excellent research to solve this problem. However, it still has some inherent challenges, such as unavoidable light variations, textureless regions, occluded areas, and nonplanar surface, that make the disparity estimation difficult [2–4].

To solve the inherent problems, numerous methods have been proposed in the past two decades. They consist of local and global methods [5, 6]. Local methods generally compute the correlation between these points and candidates over an adequate window and then use winner-takes-all (WTA) algorithm to find the best candidate to the point [7, 8]. They are fast to compute a disparity and flexible to model parametric surfaces within the neighborhood but have difficulties in handling poorly textured and ambiguous surfaces. Global methods are different from local approaches; they commonly integrate prior constraints into optimization of the point correspondences to solve the poorly textured areas and lessen the matching ambiguities. They produce the disparity map by an energy minimization algorithm and have a better performance in poorly textured and textureless regions but are limited to model piecewise planar scenes [9]. Global methods have a goodish performance when the viewpoints are close [10] but do not handle well when the space of viewpoints becomes large [11, 12].

In large-scale stereo images, ambiguous areas exist more than their short-baseline counterpart. Whether the viewpoints are close or wide, there are always some significant features, such as points of interest, which are invariable. An alternative method uses reliable feature correspondences as seeds and expands these points by using a growing-like process to obtain more point correspondences [13–18]. The methods named seed-growing or region-growing can yield much better results in large perspective distortions and increased occluded areas than traditional ones. Seed-growing methods have a low computational complexity since they are not using global optimization but are sensitive to mismatches. To lessen the influence of wrong points, Cech and Sara [19] employed an optimal solution and introduced an improved

growing method which can handle many difficult instances, such as repetitive or complex textures. The method does not need each seed to be accurate in disparity map. However, seed-growing algorithms only generate a semidense disparity map because of sparse feature points.

To overcome drawbacks of traditional matching methods and seed-growing algorithms, the matched features are naturally integrated into state-of-the-art stereo methods as soft constraints [3, 20]. In these methods, a primary work is to find accurate point correspondences as GCPs (ground control points) [21]. GCP-based approaches improve stereo matching accuracy and correctness. However, GCP-based approaches need much time to obtain an accurate disparity map.

In this paper, a two-step expansion based robust dense matching algorithm is proposed based on the previous works [19, 22–24]. Sparse support points are obtained by state-of-the-art feature matching methods [22, 23]. Before two-step expansion, the segmentation-based prior [24] is used to encode the assumption that the region which has the same color is a 3D surface. The first-step is a feature expansion that is presented based on the invariant cross ratio of projective transformation. The basic idea is to match more features from initial support points in uniform region via cross ratio constraint. However, there is no ability to find enough matched pixels to obtain dense disparity map. To obtain more point correspondences, in the second-step, the matched features from the first-step are used as seeds to grow and build a quasi-dense disparity map which is denser than the feature correspondences of the first-step but not an absolutely dense disparity. About the process stage from quasi-dense disparity to dense disparity, the paper introduces two methods: (i) fitting process: a planar surface fitting is used to remove mismatches and can fill blank occluded areas in the uniform region and (ii) synthesized method: an optimal solution incorporates quasi-dense pixels into global energy methods to reduce the matching ambiguities.

This new work mainly focuses on the first-step that uses a feature-expanded algorithm for stereo matching. In the first step, we suppose that it is a set of sparse points whose coordinates are given in the same 3D surface, and the coordinates of the homologous image pixels satisfy projective geometry of 1D in horizontal axis. Our motivation comes from the theory that the points of axis satisfy 1D projective transformation and that the cross ratio is invariant. By using the invariance of cross ratio, the inhomogeneous coordinates of each corresponding pixel can be approximated. The accurate coordinates of the corresponding pixel are found by a search model that computes a correlation statistic for neighboring pixels. In addition, to solve the poorly textured regions, we employ a propagation algorithm to expand low feature pixels. Occluded areas can be filled by a fitting process or a synthesized method, and the fitting process method does not use cross-checking (checking and optimizing the disparity by computing the differences between left-to-right disparity and right-to-left disparity). Experimental results demonstrate that the method of two-step expansion has considerable performances over the existing ones. It can produce denser disparity than these existing seed-growing algorithms, and it has a goodish result in short-baseline and wide-baseline stereo matching.

The paper is structured as follows: firstly, related work is discussed in Section 2. In Section 3, we introduce a support-point based expansion algorithm with cross ratio constraint. Then, a two-step expansion method is described, and it mainly presents the first-step about application of features expanded in Section 4. In Section 5, we describe two different methods to produce dense disparity map. Finally, we give the experimental validation supporting the feasibility of the method in Section 6. In Section 7, we give a conclusion and hint some future works.

2. Related Work

There are numerous literatures related to this work. Firstly, Scharstein and Szeliski [1] summarized dense stereo methods and established an early test bed for stereo matching algorithm. Then, Geiger et al. provided a newly outdoor challenge [25] for the quantitative evaluation of large-scale stereo matching. Seitz et al. [26] introduced a comprehensive study and made a comparison of stereo techniques. It included two main strategies for obtaining stereo correspondence: feature correspondences based local approaches and energy-minimum based global methods. In our method, the previous two-step expansion algorithm and the latter fitting process stage belong to the first strategy, and the later synthesized method falls in the second one.

Dense energy-minimum based global methods had a good performance in the past decade. Local stereo algorithms based on feature correspondences are speedy to estimate disparity [1, 27] but cannot effectively handle the blurry border and mismatches [7]. Hence, most excellent stereo matching algorithms rely first on using local approaches to find the pixel correspondences and then incorporate them into global constrains by dynamic programming (DP) [28–31], level sets [32], space carving [33], PDE [12, 34], EM [35], and voxel coloring [36]. Recently, two global methods based on Markov random fields (MRFs) are used as basic algorithms to be improved: Graph Cuts [37] and Belief Propagation [38]. Many works of research about both of them have achieved a desirable result [4, 39, 40]. Both methods are often used to be comparable data of the top contenders in the realm of dense stereo matching and are powerful tools to produce disparity map but intractable to finish the solution in wide-baseline stereo. In contrast, our method can lessen the matching ambiguities and is efficient to large-scale stereo matching.

Sparse local feature based approaches are robust to the large-scale images. Image features play an important role in computer vision. They have already been used in wide-baseline stereo matching [41–43]. In a wide-baseline setup, the inherent problems are perspective distortions and occlusion. Feature based matching methods are particularly effective because features are robust, distinctive, and invariant to various image and scene transformations [22, 23, 44–47]. However, the traditional methods based on feature matching produces only sparse pixel correspondences. To find more

matched points than features, a propagation algorithm from the matched points to their neighbors is introduced.

The rule of growing a region from primary seeds was used to segment image [48]. The seed-growing principle was originally introduced into stereo matching by Otto and Chau [49], O'Neill and Denos [50], and Kim and Muller [51] and used for photogrammetric community. Then Lhuillier and Quan [15, 52] employed the epipolar constraint and uniqueness constraint to greedily reproduce adjacent components in disparity blankness from corresponding seeds. The growth algorithm cannot achieve a good performance in the areas of repetitive patterns. The best first strategy as an optimal solution was used to replace the pixel-wise growth increments by Zeng et al. [17, 18]. And the optimization cannot be able to remove the previous match errors, especially in complex scenes. Kannala and Brandt [53] and Megyesi et al. [54] introduced a propagation algorithm by affine deformation of image similarity patches. But it had inaccurate affine parameters due to wrong initial seeds and made a bad propagation. Cech and Sara [19] introduced an optimal solution and presented a seed-growing method that could recover from errors in initial seeds. However, the method only produced a semidense disparity map. In contrast, our method can not only handle the difficult instances (e.g., repetitive texture, complex scene, and wrong initial seeds) but also produce denser point correspondences than the existing methods.

To compute an accurate dense disparity map, we incorporate quasi-dense pixel correspondences as GCPs into state-of-the-art global matching framework. In these literatures about stereo matching, GCPs-based methods can achieve a precise result. Bobick and Intille [2] used GCPs to optimize DP solution and reduce large occlusion. GCPs were used in preprocessing stage to guide the previous matching process and could reduce false matched points by using the method of Kim [3] and Wang et al. [20]. In [21], a GCPs-based regularization was incorporated into global method by using the Bayes optimization rule. In contrast, our method does not need provided special GCPs and can offer quasi-dense pixel correspondences as GCPs.

Geiger et al. proposed a generative probabilistic model ELAS [7] for wide-baseline stereo matching and offered a challenging KITTI dataset [25]. On KITTI dataset, these methods [55–57] that were used to compute optical flow had better results. In contrast, our method is a just strategy for stereo matching and receives a result compared with ELAS.

3. Efficient Expansion with Cross Ratio Constraint

3.1. Cross Ratio Constraint Model. In the epipolar geometry of two views, it can restrict the corresponding point on the polar line. To find the precise position of the corresponding point, traditional algorithms employ exhaustive search along the corresponding line and give a statistic for correlation of all candidates. To fasten the position estimation of the corresponding point on line, we introduce a new constraint based on 1D projective geometry.

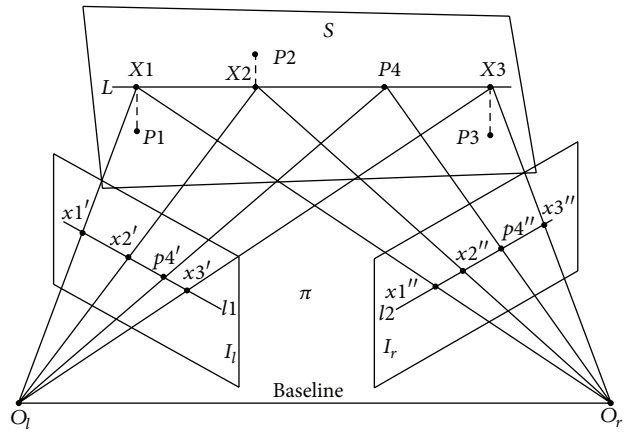


Figure 1: Two cameras are indicated by their centres O_l and O_r and their image planes I_l and I_r. There are 4 3D points $P1$, $P2$, $P3$, and $P4$ in uniform 3D planar surface S, and the point $P4$ projects to $p4'$ and $p4''$ in the images I_l and I_r, respectively. Line $l2$ in the right image and line $l1$ in the left image are epipolar lines separately with respect to points $p4'$ and $p4''$. Two camera centres, 3-space point $P4$, and its images $p4'$ and $p4''$ lie in an epipolar plane π. The intersection of the planes π and S determines the line L in 3D. The 3D points $X1$, $X2$, and $X3$ are the closest points on the line L to the 3D points $P1$, $P2$, and $P3$. Points $x1'$, $x2'$, and $x3'$ in the left image and points $x1''$, $x2''$, and $x3''$ in the right image are projected by 3D points $X1$, $X2$, and $X3$, and points $x1'$, $x2'$, and $x3'$ lie on the line $l1$, and points $x1''$, $x2''$, and $x3''$ lie on the line $l2$.

We assume there is a stereovision system as shown in Figure 1. It can be seen that there are three sets of four collinear points in the polar plane π. Each set is related to the others by a line-to-line projective transformation. Since the cross ratio is invariant under a 1D projective geometry, it has the same value as

$$\frac{X1P4}{X2P4}\frac{X2X3}{X1X3} = \frac{x1'p4'}{x2'p4'}\frac{x2'x3'}{x1'x3'} = \frac{x1''p4''}{x2''p4''}\frac{x2''x3''}{x1''x3''}. \quad (1)$$

3.2. Estimation Model via Cross Ratio. The cross ratio constraint based on 1D projective geometry needs three or more known 3D points. Thus, it needs to obtain the known 3D points. We employ features matching algorithm as prior to produce reliable point correspondences which can be used to calculate the fundamental matrix. The proportional coordinates of the known 3D points can be estimated by the point correspondences and the fundamental matrix. It can produce more errors when the region including the known 3D points is not a planar surface.

To lessen the error, we know that the points on the same epipolar line satisfy 1D projective geometry whether the surface is plane or not and introduce a search strategy that uses image points near the epipolar line instead of the 3D points, as shown in Figure 2. Suppose the images have been rectified and the point correspondences lie on the same line in both images. We wonder how the corresponding point $p4''$ in the right image is found. Firstly, we can find the

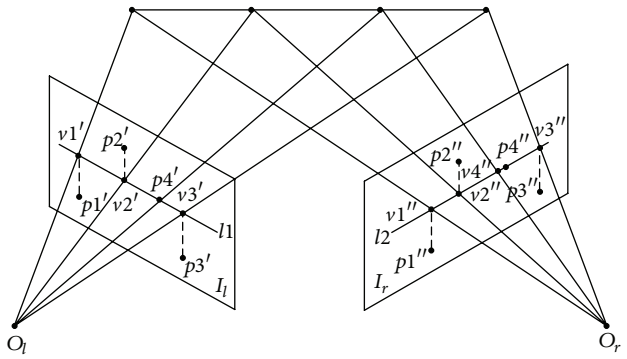


FIGURE 2: I_l and I_r are the images observed from the camera centres O_l and O_r separately. Given three sets of matched point pairs($p1'$, $p1''$), ($p2'$, $p2''$), and ($p3'$, $p3''$), an unmatched point $p4'$ in the left image and the unknown corresponding point $p4''$ in the right image, $l1$ and $l2$ are the lines on which the corresponding points $p4'$ and $p4''$ lie. Points $v1'$, $v2'$, and $v3'$ are the closest points on the line $l1$ to the points $p1'$, $p2'$, and $p3'$ in the left image, and points $v1''$, $v2''$, and $v3''$ are the closest points on the line $l2$ to the points $p1''$, $p2''$, and $p3''$ in the right image.

corresponding polar lines $l1$ and $l2$. Then we can find a point $v4''$ satisfies the following equation:

$$\frac{v1'p4'}{v2'p4'}\frac{v2'v3'}{v1'v3'} = \frac{v1''v4''}{v2''v4''}\frac{v2''v3''}{v1''v3''}. \quad (2)$$

If the points $p1'$, $p2'$, $p3'$ and $p1''$, $p2''$, $p3''$ are the homologous points that the 3D points $P1$, $P2$, and $P3$ project to two images separately in Figure 1, the points $v1'$, $v2'$, $v3'$, and $v1''$, $v2''$, $v3''$ are not the corresponding points projected by the 3D points $X1$, $X2$, and $X3$ because of projective transformation. Hence, the point $v4''$ is not the corresponding point $p4''$, but adjacent to the point $p4''$. The distance points to the epipolar line are shorter; the position of $v4''$ is nearer to the point $p4''$. We employ a probabilistic search strategy to ensure the point $p4''$ in the $v4''$ has contiguous pixels along the line $l2$.

3.3. Search Strategy. The search strategy computes all correlations with the neighbors of the point $v4''$ and decides the position of the point $p4''$. As shown in Figure 3, a set of neighborhoods $\{c(1), \ldots, c(m), v4'', c(m+1), \ldots, c(2m)\}$ of size $(2 \times m + 1)$ whose centre is point $v4''$ is built as a set of candidate matches. The value m is the radius of search and is decided by the maximum d of the Euclidean distances from referenced points $p1''$, $p2''$, and $p3''$ to the line $l2$. If the distance from reference point pi'' $(i = 1, 2, 3)$ to the line $l2$ is di, then the maximum is $d = \max(di)$ and $m = \min(\text{ratio} \times d, r)$, where ratio, r, is nonzero constants for proportional d and fixed radius. We use sum of absolute differences (SAD) [58] on $N \times N$ window as image similarity statistic between the point $p4'$ and all candidate points in the right image, where N is a positive constant. Assuming the SAD value between the point $p4'$ and candidate points is $D(i)$, where $i = 1, 2, 3, \ldots, 2 \times m + 1$, if the SAD value between $p4'$ and $v4''$ is $Dv = D(m+1)$, and the nth point in candidate ones except the point $v4''$ has the SAD minimum and $D(n) = \min\{D(i)\}$, where $i = 1, 2, \ldots, m, m+2, \ldots, 2 \times m + 1$, then the corresponding point $p4''$ is defined as

$$p4'' = \begin{cases} v4'', & \text{if : } Dv < D\min \times \alpha, Dv < \beta, \\ c(n), & \text{if : } Dv > D\min \times \alpha, D\min < \beta, \\ -1, & \text{if : } Dv > \beta, D\min > \beta, \end{cases} \quad (3)$$

where α is a proportional constant and β is the threshold for the correct correspondence, if $p4'' = -1$ means no corresponding point.

4. A Two-Step Expansion Method

In this section, we describe a two-step expansion algorithm based on image features to compute quasi-dense point correspondences between two views. Our method is inspired by observing an instance where all points in the uniform surface satisfy 1D projective geometry in horizontal axis. And in 1D projective transformation, the cross ratio of the projected points is invariant. Our algorithm is arranged as follows: firstly, a sparse set of initial support points are found by excellent feature matching method. Then, in the first-step expansion, we use segmentation-based prior to partition the image into different regions and employ the invariance of cross ratio as a restrictive condition to find the more corresponding feature points from the support points in the same region. Finally, a regular seed-growing approach is used to obtain more pixel correspondences as the second-step expansion.

Suppose there exists a pair of images $I = \{I_l, I_r\}$, where I_l, I_r are the separately left and right images, this section aims at finding the quasi-dense disparity D_q corresponding to I_l. To expediently introduce our method, we suppose input images I_l and I_r are rectified, such that corresponding points lie on epipolar lines of two images.

4.1. Initial Support Points. Before expansion, we introduce how to establish a sparse set of feature correspondences as initial support points. Most algorithms which are used to extract image features can be categorized as either corner detectors (such as Harris and Stephens [22] and SUSAN [46]) or descriptor extraction (such as SIFT [23], SURF [44], and DAISY [47]). Recently, a regional feature detector [59] based on descriptor [23] had a good performance in dealing with the large-scale instance. In our method, we employed regular Harris method [22] to obtain initial support points. While in the presence of large disparity ranges, the number of the successfully matched Harris points is less than the threshold Np (which is decided by the number of the segmented regions; refer to Section 4.2), scale invariant feature transform (SIFT) algorithm is used to extract features, and the KD-tree with the best bin first (BBF) [60] algorithm is employed to index and match these features. We assume the $\{(pi', pi'')\}_i$ is matched point pairs by feature matching method, where $\{pi'\}$ and $\{pi''\}$ are points from two images.

4.2. The First-Step Expansion. At this stage, our objective is to compute all the possible feature point correspondences

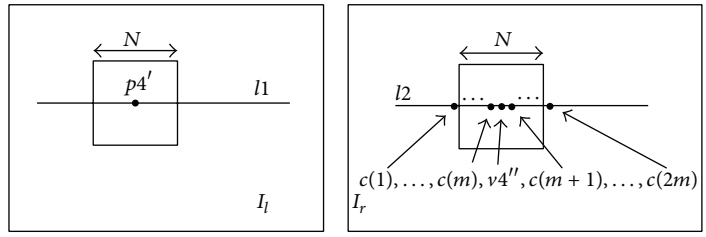


Figure 3: I_l and I_r and $l1$ and $l2$ are separately the corresponding planes and lines. Given a pair of points $(p4', v4'')$ whose $v4''$ is estimated by cross ratio model, $\{c(1), \ldots, c(m), v4'', c(m+1), \ldots, c(2m)\}$ is a set of neighborhoods of the point $v4''$. N is the size of window which is used to compute correlation.

through the initial support points in the uniform region. The first-step expansion is based on segmented regions; thus, we employ the mean-shift method to segment the reference image I_l before expanding feature points. The mean-shift algorithm which was successfully used to partition images by Comaniciu and Meer [24] can ensure our method estimates regions correctly and localizes depth boundaries precisely. The result of image segmentation will set a different label to each segmented region. If the number of the segmented regions is Kl, then the threshold Np in Section 4.1 is $Np = 3 \times Kl$.

In Section 3, we introduce an expansion model based on 1D projective geometry in the same planar surface. We now use the expansion model as the first-step expansion algorithm. More formally, let $L = \{l_1, l_2, l_3, \ldots, l_k\}$ be a set of labels with respect to the different segmented regions of the left image. Each pixel pi is assigned to a corresponding label l_j where $l_j \in L$.

We assume the initial support points belonging to a label l_j construct a set of samples $Sa_j = \{(pi', pi'')\}_i$, where $pi' \in l_j$. In this step, we spread sweeping feature correspondences from initial support points in the same region. In our method, the feature means a point whose absolute value of gradient is bigger than 1. Hence, our prior is a process that computes gradient $G(x, y)$ of each pixel in the image and selects the pixels whose gradient $|G(x, y)| \geq 1$ as our candidate feature points. Suppose we have found all the feature points $\{xi'\}$ and each feature $xi' \in \{xi'\}$ is assigned to a corresponding label l_j. We will introduce how to find the corresponding point xi'' through Sa_j.

The expansion algorithm mainly is based on epipolar geometry and 1D projective transformation. Epipolar restraint has been used to rectify the image and restrict the corresponding points to the same lines in the images. We just need to find three support points to estimate the probabilistic position of the corresponding point. The number of the initial support points $\{pi\}$ in each region l_j is not fixed and can be sorted to two statuses: more than three points and no more than three points. This step mainly handles the first status.

When the number of $\{pi\}$ is more than three, as shown in Figure 4, we can consider each horizontal axis on which pixels lie is its corresponding epipolar because of image rectification. Suppose each point $pi' = (xi', yi')$, where $i = 1, 2, \ldots, 8$, and three support points which would be found satisfy the following conditions: (i) three points have a minimum summation of the distances to the epipolar line

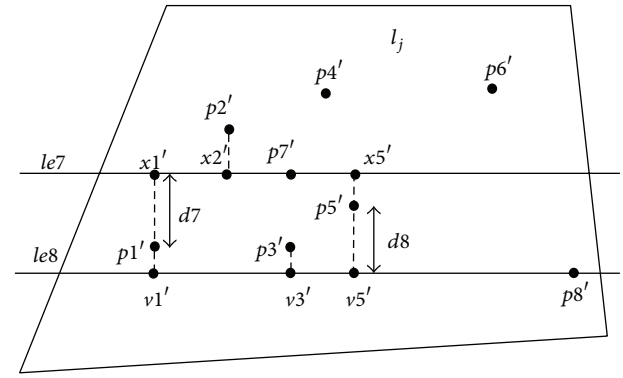


Figure 4: In the region l_j, a set of matched points $\{p1', p2', p3', p4', p5', p6'\}$ has been known, and $p7'$ and $p8'$ are the points which need to search the corresponding point. The horizontal axes $le7$ and $le8$ are the epipolar lines of the points $p7'$ and $p8'$, separately.

at the same time and (ii) any two x-coordinates of all points should not be equal. For example, as shown in Figure 4, the support points of $p7'$ are $p1'$, $p2'$, and $p5'$ and the support points of $p8'$ are $p1'$, $p3'$, and $p5'$, that satisfy the two conditions. The corresponding search radii are separately $m7 = \min(\text{ratio} \times d7, r)$ and $m8 = \min(\text{ratio} \times d8, r)$. Then the corresponding point can be found by the method of Section 3.

4.3. The Second-Step Expansion. The second-step employs a regular seed-growing method to obtain stable correspondences in poorly textured regions. Suppose the first-step produces a list of point correspondences $S = \{(pi', pi'')\}_i$. We regard the point correspondences S as seeds to grow corresponding patches. Despite the fact that the first- step can find more effective point correspondences, it inevitably introduces errors in complex areas. The traditional seed-growing algorithms do not handle wrong initial seeds well. To overcome the drawbacks, Cech and Sara [19] temporarily forwent uniqueness constraint, propagated most disparity ingredients, and then optimized them to remove these false disparity components. Hence, the second-step employs the method of Cech to obtain quasi-dense disparity D_q.

Cech method includes two phases: (i) growing and propagating as many seeds as possible regardless of their overlaps and (ii) optimizing these seeds of the first phase and

removing these false ones. The seed-growing method of Cech can keep accurate point correspondences and recover most disparities from false seeds. The detailed descriptions of the seed-growing method can be referred to the literature [19].

5. Obtaining Dense Disparity Map

The two-step expansion method cannot find all pixel correspondences in some regions because of occlusion and cannot produce completely dense disparity map. We introduce two different processes to compute dense disparity map from quasi-dense point correspondences. One is a filling process by regional 3D surface fitting; the other is a synthesized method that integrates quasi-dense pixel correspondences as GCPs into global optimization frameworks in a principled way.

5.1. Fitting Process. In Section 4.2, we have obtained different regions from the image I_l by the segmentation-based prior. The segmented regions may be with respect to different 3D surface. Now we assume each 3D surface is planar. In some regions of a quasi-dense disparity D_q, there may be only a few corresponding points and some piecewise patches which are built by unmatched points due to occlusion. A 3D planar surface fitting can be applied to fill the uncharted patches in the same region.

Assume there exists a set of pixel correspondences $\{(pi', pi'')\}_i$ in an arbitrary region l_j, and we use the regional data $\{pi'\}$ to compute a 3D plane Pl_j. We describe each pixel of the quasi-dense disparity D_q as $V_i = (xi', yi', zi')$, where xi' and yi' are the coordinates of pi' in the image, and zi' is a corresponding disparity. Then, we can use a set of points $\{V_i\} \in Pl_j$ to fit a 3D planar surface:

$$f\left(xi', yi', zi'\right) = axi' + byi' + czi' + d, \tag{4}$$

where a, b, c, and d are the parameters which are used to describe a plane. These pixels belonging to the same area satisfy the 3D plane equation and can be computed the involved disparities.

5.2. Synthesized Method. Recently, a mixed stereo model which uses these known point correspondences as the GCPs to improve the result of global matching has a good performance in textureless areas and occlusion.

Synthesized method is inspired from the method of Wang [21] and formulates the stereo modal as a MAP-MRF problem. Assume the quasi-dense D_q is produced from a pair of images $I = \{I_l, I_r\}$ by two-step expansion. Based on Bayes' rule, the posterior probability of the disparity map D is expressed as $P(D \mid I, D_q) \propto P(I \mid D)P(D_q \mid D)P(D)$. Finding the maximum posterior cost means minimizing the corresponding negative log likelihood. Thus computing a disparity map D becomes the problem for minimizing the energy function:

$$E(D) = E_{\text{data}}(D) + E_{\text{smoothness}}(D) + E_{\text{quasi-dense}}(D), \tag{5}$$

where E_{data} is a function to estimate the probability for disparity map, $E_{\text{smoothness}}$ is a smoothness term to encourage similar neighboring points in locally smooth region, and $E_{\text{quasi-dense}}$ is the energy of D_q to constrain the accuracy of disparity map D. The details can be referred to the literature [21].

Table 1: Comparison of the results on number of corresponding points.

	St. Martin	Head	Larch
Cech 07	481733	293004	165218
Our method	749585	379153	195531

5.3. The Overall Algorithm. The process of two-step expansion algorithm is summarized as in Algorithm 1.

6. Experiments

We took different experiments to demonstrate the validity of our method. In Section 6.1, we compared our approach to the seed-growing method of Cech on the real complex scenes [19]. It was a test on running time for different image resolutions in Section 6.2. We then separately evaluated fitting process and synthesized method on Middlebury benchmark short-baseline stereo images with known ground truth data in Section 6.3. In Section 6.4, we tested our algorithm on large-scale stereo image pairs.

Throughout all experiments we set ratio $= 0.3$, $r = 7$, $N = 5$, $\alpha = 1.2$, and $\beta = 4 \times 10^{-4}$, which were empirically determined. All experiments were operated on the computer with Intel core 2 duo CPU and 2.93 GHz clock frequency. Unless stated otherwise, we employed regular Harris method to obtain initial matched points and performed mean-shift image segmentation using EDISON code [61] implementation of Comaniciu's paper [24].

6.1. Computing Quasi-Dense Disparities. Firstly, we can obtain quasi-dense disparity map by the two-step spreading. We demonstrated the difference between the seed-growing method and our algorithm by comparing their performances on some real data. In known seed-growing algorithms, the method proposed by Cech and Sara [19] has a better performance, even in the presence of repetitive patterns. Hence, we compared our approach to the method of Cech. We tested different stereo images from the Cech dataset [62], that is, St. Martin, Head, and Larch. The relevant quasi-dense disparities of the images are shown in Figure 5. It can be seen that our algorithm can produce denser disparity map than the algorithm proposed by Cech. A comparison to the number of the corresponding points in different images is depicted in Table 1.

This experiment result demonstrates that our method can produce a quasi-dense disparity map via a sparse set of initial feature correspondences. Our method does not need too accurate matched features as seeds. In a repeated experiment, our method always found more point correspondences than Cech's method.

Input: A pair of rectified images I_l and I_r from different viewpoints of one scene;
Set the values of *ratio*, r, N, α, and β.
Output: The disparity map with respect to I_l.
Begin:
Step 1. Finding initial point correspondences $\{(pi', pi'')\}_i$ by using state-of-the-art matching method;
Step 2. Using the mean-shift to partition the image I_1 to different areas denoted as: $L = \{l_1, l_2, l_3, \ldots, l_k\}$;
Step 3. Assigning the points $\{pi'\}$ into the corresponding area l_j;
Step 4. Removing the coarse mismatches in the area l_j by using regional affine transformation;
Step 5. Computes gradient $G(x, y)$ of each pixel in the image and selects the pixels $\{xi'\}$ whose gradient $|G(x, y)| \geq 1$ as our candidate feature points;
Step 6. Assign each feature xi' to a corresponding label l_j;
Step 7. The first step for the matched feature expansion
Repeat:
for: $i = 1$: size($\{xi'\}$)
ensure l_j via to: $xi' \in l_j$;
find a set of samples $Sa_j = \{(pi', pi'')\}_i$, where $pi' \in l_j$;
if: size($\{p_i\}$) > 3
compute the point xi'' by estimation model via cross ratio;
end if
end for
Until $\{l_j\}$ is empty
Step 8. The second step expansion by using a regular seed-growing method
Step 9. Obtain dense disparity map by using fitting process or synthesized method

Algorithm 1: Two-step expansion algorithm.

6.2. Running Time. The term of running time is relative to the elements, that is, image resolution, segmented regions, and initial support points. We changed the image resolutions for Tsukuba, Teddy, Cones, and Venus from Middlebury benchmark [63] and then took a running time statistic with respect to the three elements separately. We downscaled the images bicubically by 10%~90%, tested the running time in different resolutions, and recorded the corresponding numbers of regions and points. As shown in Figures 6(a), 6(b), and 6(c), it is related to the illustration about the running time of images in different resolutions, regions, and points. It can be seen that the numbers of the segmented regions and the initial points are more, and the corresponding running time is shorter in the same resolution. Figure 6(d) shows the corresponding relations between segmented regions and matched points in different image resolutions.

6.3. Short-Baseline Stereo Matching. We tested the fitting process and the synthesized method on several image pairs, that is, Tsukuba, Venus, Teddy, and Cones from the Middlebury benchmark [63]. The maximum of the disparity for the images is less than 100. Firstly, we used a two-step method to produce the different quasi-dense disparities of the images. Then, we found the corresponding disparity maps by the fitting process and synthesized method. When we implemented the fitting process, we restricted the maximal difference of disparity in the same region less than 10. In the synthesized method, we employed Graph Cuts [37] as our assistant global method. The goal is to compute a disparity map D by the function (5). In time statistics, the fitting process takes about 1.3 minutes to estimate a disparity and the synthesized method takes about 1.8 minutes. We demonstrate in Figure 7 the results for the images of Tsukuba, Venus, Teddy, and Cones. As can be seen, these disparities produced by synthesized method have a clear structure and few blurry areas.

To evaluate the performance of our method, we used the quality measure method proposed in [1] with known ground truth data to evaluate the synthesized results. The matching results rank 87 and 62 with respect to 1 and 0.5 pixels error in Middlebury website. These competitive algorithms were based on these classical original methods and proposed an improved algorithm. They commonly integrated many technologies into their methods and had a better performance. Our method is first proposed by us without more refined technologies. To verify the validity of the method, we compared it with these classical original methods, that is, GC (graph cuts) [37], CSBP (constant-space belief propagation) [64], DP [29], and SO (scanline optimization) [1], as shown in Table 2, where the absolute error is more than 2 pixels. Quality evaluation uses the three performance measures: nonocc (bad pixels of nonoccluded), all (bad pixels of the entire image), and disc (bad pixels of near discontinuous).

It can be seen from Table 2 that our method has a better result than these of the traditional methods. The wider the baseline is, the more obvious the accuracy of the disparities is, for example, the scenes of Teddy and Cones. Because our method is based on feature matching which is efficient to large-scale images. On the Venus scene, the error of our method is not the lowest in these methods. Since global methods employ energy-minimum function to optimize the disparity map and have a better performance than feature-based local methods in short-baseline. The images of Venus scene belong to short-baseline. The global methods of GC

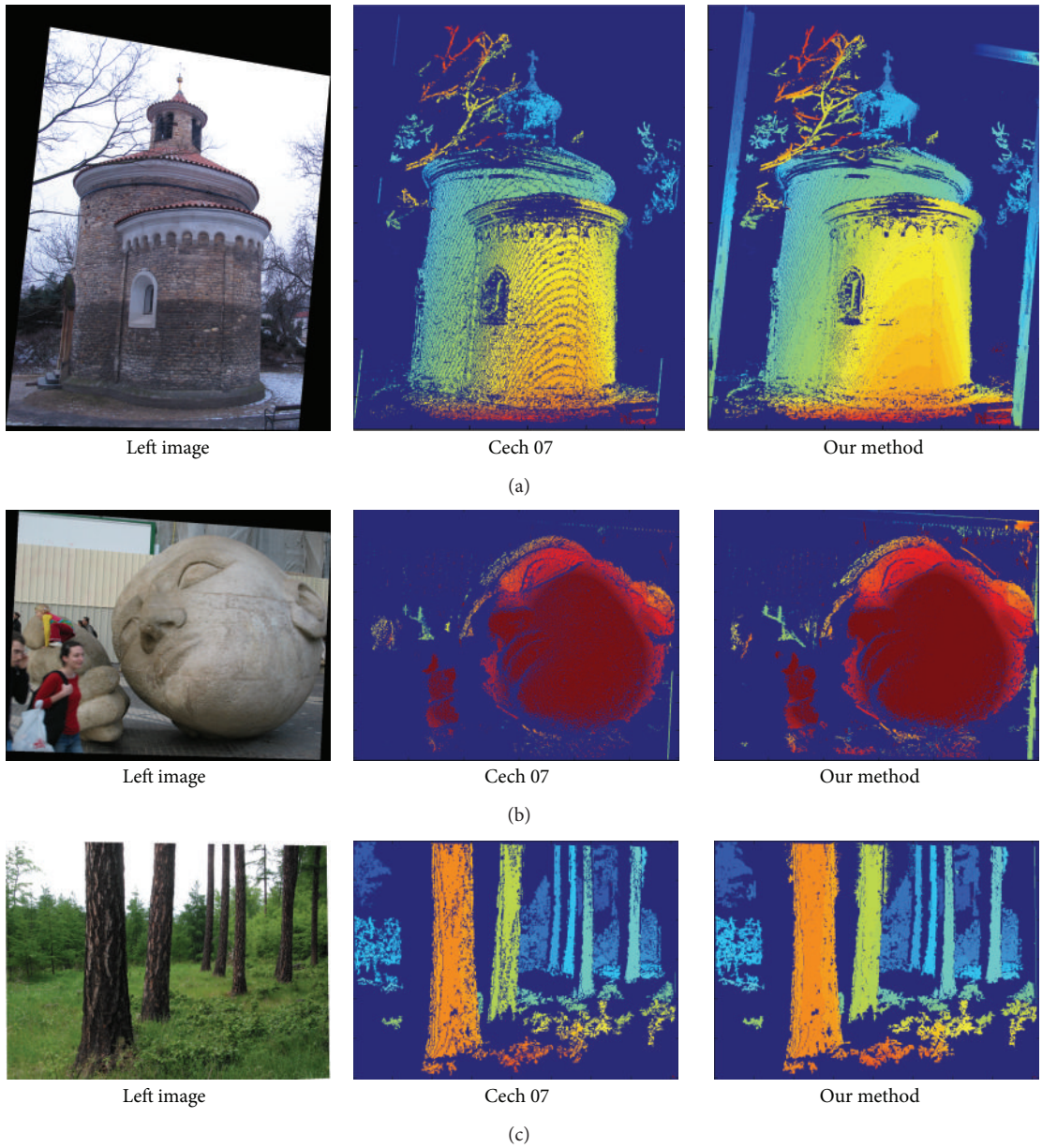


Figure 5: Results for quasi-dense disparities of Cech dataset are as follows: (a) St. Martin, (b) Head, and (c) Larch. Disparity maps are partitioned in different colors: colder color means smaller disparities, warmer color means larger disparities, and deeply blue areas are unassigned disparity.

Table 2: Comparative performance of stereo algorithms according to Middlebury methodology.

	Tsukuba			Venus			Teddy			Cones		
	Nonocc	All	Disc	Nonocc	All	Disc	Nonocc	All	Disc	Nonocc	All	Disc
Ours	2.57	3.13	10.8	3.34	3.49	13.1	5.44	7.84	12.1	3.41	8.30	9.21
GC	1.67	3.75	8.20	0.83	2.37	8.12	9.72	18.8	17.2	4.51	15.0	11.2
CSBP	1.74	3.84	9.10	1.09	2.52	12.9	8.18	17.3	20.6	4.07	14.2	11.3
DP	3.43	4.23	9.85	6.50	7.43	17.4	7.11	14.9	13.4	6.52	15.1	15.1
SO	4.23	6.21	10.7	5.14	6.58	17.0	11.3	20.2	18.6	8.76	18.8	16.1

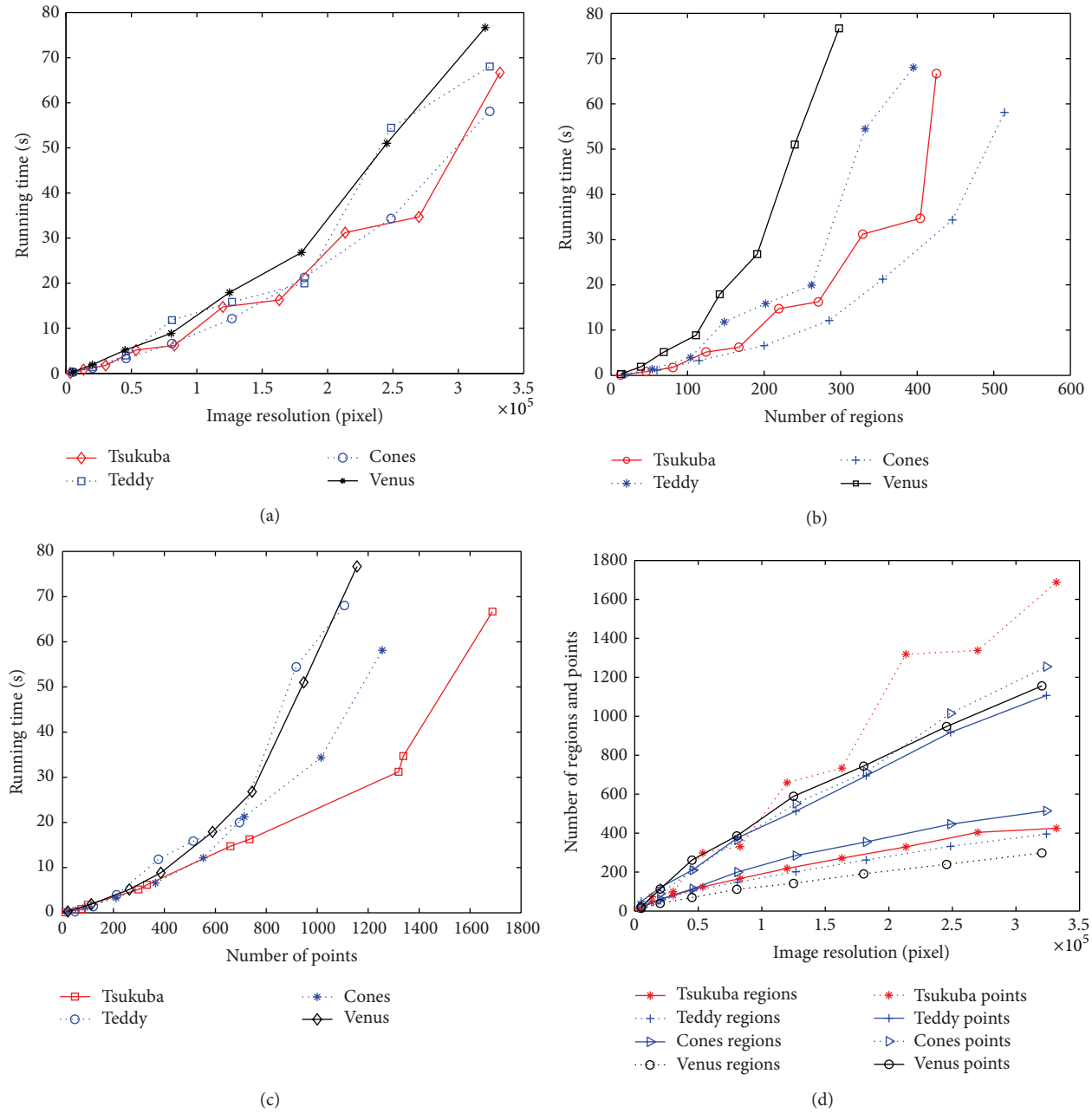


FIGURE 6: The relations of running time to (a) image resolution, (b) number of regions, (c) number of support points on the Tsukuba, Teddy, Cones, and Venus image pairs, and (d) the relevant segmented regions and corresponding points in different resolutions of the images.

and CSBP have more accurate results than our method on the Venus scene.

6.4. Large-Scale Stereo Matching. Though short-baseline stereo matching can yield accurately dense disparity, there is much more challenge in large-scale stereo images because of too much occlusion. In large-scale stereo images, we computed the disparity just via the fitting process and without the synthesized method. Firstly, we compared the fitting process on a wide range of baseline high resolution images, that is, Aloe and Raindeer from the Middlebury benchmark [63] whose maximum of the disparity is bigger than 200. In particular, we compared our method against the method ELAS proposed by Geiger et al. [7], as shown in Figure 8. We compute all erroneous pixels of the entire image whose absolute error is more than 3 pixels. The error results of our method are 13.03% and 20.36% separately corresponding to the images Aloe and Raindeer, and the results of ELAS are 14.14% and 22.28%.

Then, we took a test on the KITTI dataset [25], which consists of 194 training and 195 test pairs of urban images. The training images with semidense ground truth disparities are used to adapt the parameters of stereo matching methods.

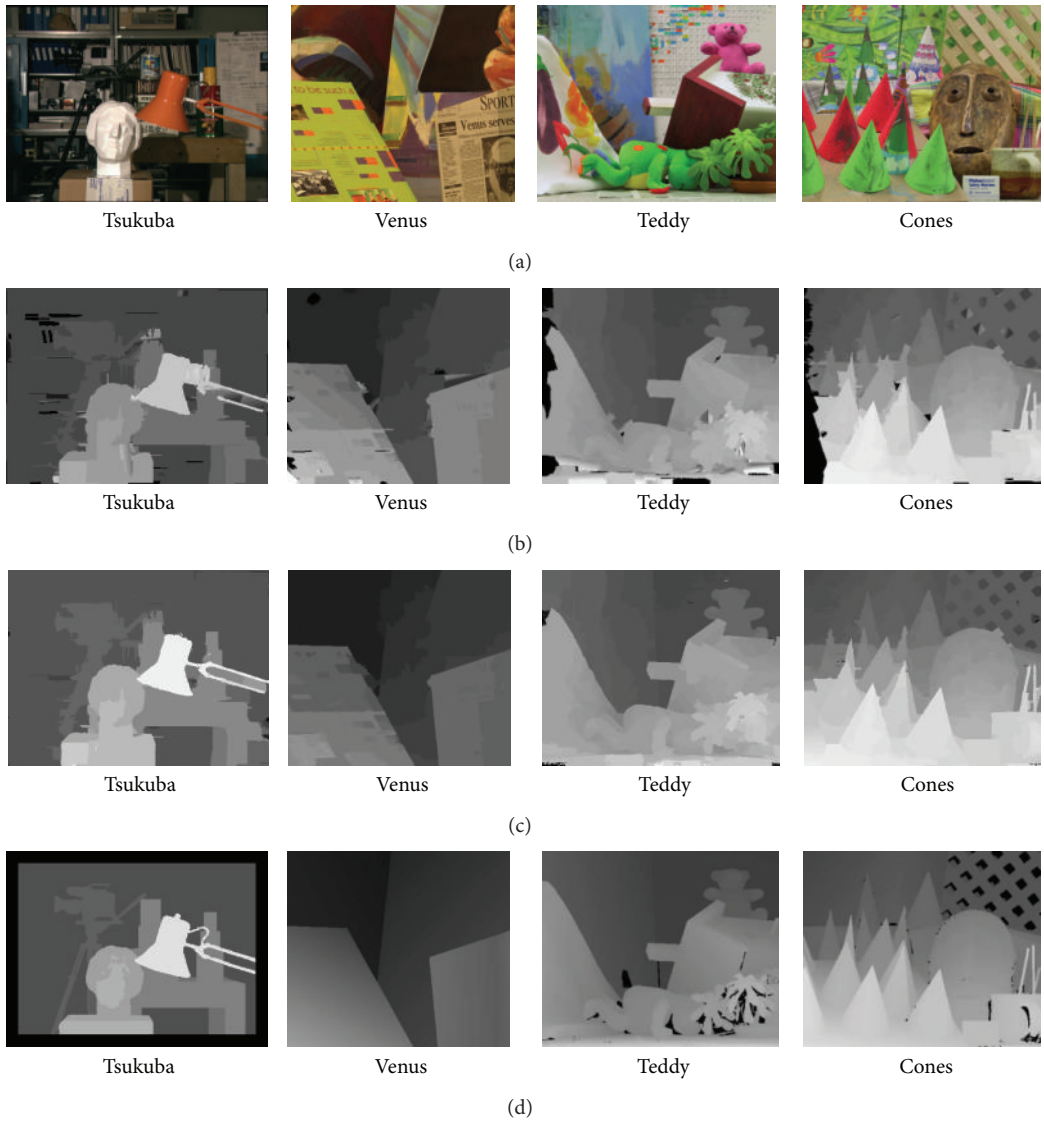


Figure 7: Results of our two different methods on short-baseline dataset. (a) Left images. (b) Results of fitting process. (c) Results of synthesized method. (d) Ground truth disparities.

There is no parameter to be trained and modified in our method. The test images without ground truth are used to evaluate participants in the challenge. On the dataset, the main problem is how to handle these textureless areas. We computed the disparity maps of these test images via the fitting process, and some results are shown in Figure 9. The average run time for computing a disparity map is about 4.7 minutes. The matching results rank 38 and 35 with respect to 3- and 5-pixel error in KITTI website. We compared our method to the similar methods, that is, ELAS [7], GCSF (growing correspondence seeds flow) [55], and GCS (growing correspondence seeds) [19], as shown in Table 3, where Out-Noc is the percentage of erroneous pixels in nonoccluded areas, and Out-All is the percentage of erroneous pixels in total. Avg-Noc is the ratio of average disparity or end-point error in nonoccluded areas. Avg-All is the ratio of average disparity or end-point error in total. The qualitative results for this dataset are similar to the previous evaluation. We are able to robustly reconstruct large-scale images, which leads to low error rates on the street and on other slanted surfaces.

7. Conclusion

In this paper, we introduce a two-step expansion to produce precise disparity maps from stereo images whether the stereo baseline is short or large. Our method is based on feature matching and can cope with the difficult cases such as large perspective distortions, increased occluded areas, and complex scenes. Our experiments on Cech's dataset, the Middlebury benchmark, and KITTI dataset demonstrate that our method achieves good results in the real complex

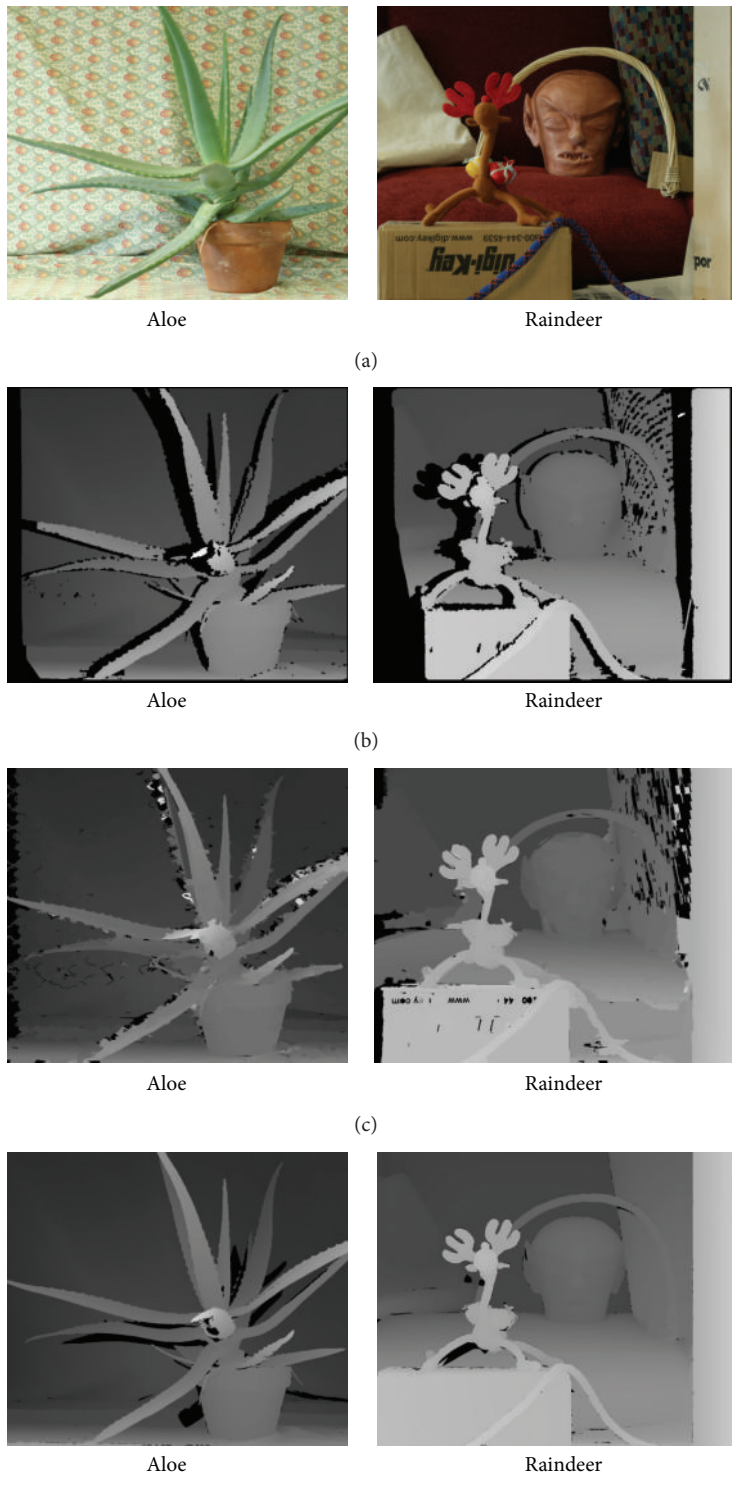
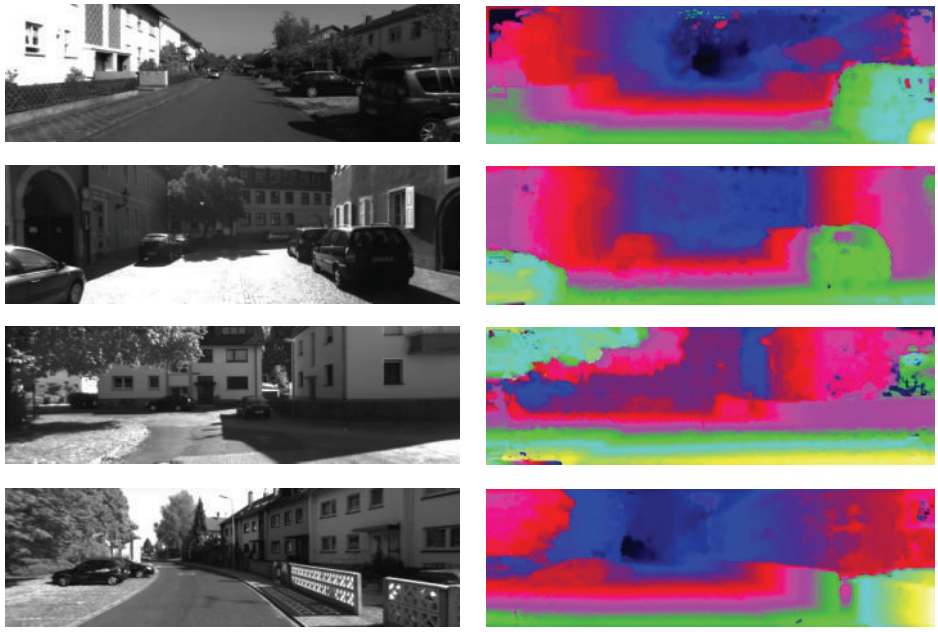


Figure 8: Comparison to Geiger's method on the Aloe and Raindeer image pairs. (a) Left images. (b) ELAS results. (c) Fitting process results. (d) Ground truth disparities.



(a) (b)

Figure 9: Results on urban scenes. (a) Left images. (b) Our method results. Best viewed is in different colors.

Table 3: Comparative evaluation results on KITTI test dataset.

	>2 pixels		>3 pixels		>4 pixels		>5 pixels		End-point	
	Out-Noc	Out-All	Out-Noc	Out-All	Out-Noc	Out-All	Out-Noc	Out-All	Avg-Noc	Avg-All
ELAS	10.96%	12.83%	8.24%	9.96%	6.73%	8.24%	5.67%	6.97%	1.4 px	1.6 px
Ours	14.59%	16.08%	9.91%	11.30%	7.32%	8.57%	5.74%	6.87%	1.7 px	1.9 px
GCSF	17.41%	18.73%	12.05%	13.24%	9.22%	10.28%	7.54%	8.49%	1.9 px	2.1 px
GCS	19.03%	20.32%	13.38%	14.54%	10.41%	11.43%	8.64%	9.55%	2.1 px	2.3 px

scenes, short or wide baseline image pairs. Importantly, we introduce a cross ratio restraint model to expand more feature correspondences based on state-of-the-art feature matching.

Our method primarily involves performing point computation in large numbers of segmented regions, which is fit for implementing in GPU and can real-time compute the disparity map of stereo images.

Conflict of Interests

The authors declare that there is no conflict of interests regarding the publication of this paper.

References

[1] D. Scharstein and R. Szeliski, "A taxonomy and evaluation of dense two-frame stereo correspondence algorithms," *International Journal of Computer Vision*, vol. 47, no. 1–3, pp. 7–42, 2002.

[2] A. F. Bobick and S. S. Intille, "Large occlusion stereo," *International Journal of Computer Vision*, vol. 33, no. 3, pp. 181–200, 1999.

[3] J. C. Kim, K. M. Lee, B. T. Choi, and S. U. Lee, "A dense stereo matching using two-pass dynamic programming with generalized ground control points," in *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pp. 1075–1082, June 2005.

[4] J. Sun, N. Zheng, and H. Shum, "Stereo matching using belief propagation," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 25, no. 7, pp. 787–800, 2003.

[5] H. Sadeghi, P. Moallem, and S. A. Monadjemi, "Feature based dense stereo matching using dynamic programming and color," *International Journal of Computational Intelligence*, vol. 4, no. 3, p. 179, 2008.

[6] L. Valgaerts, A. Bruhn, M. Mainberger, and J. Weickert, "Dense versus sparse approaches for estimating the fundamental matrix," *International Journal of Computer Vision*, vol. 96, no. 2, pp. 212–234, 2012.

[7] A. Geiger, M. Roser, and R. Urtasun, "Efficient large-scale stereo matching," in *Proceedings of the 10th Asian Conference on Computer Vision (ACCV '10)*, November 2010.

[8] B. M. Smith, L. Zhang, and H. Jin, "Stereo matching with non-parametric smoothness priors in feature space," in *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR '09)*, pp. 485–492, Miami, Fla, USA, June 2009.

[9] L. Tang, H. T. Tsui, and C. K. Wu, "Dense stereo matching based on propagation with a Voronoi diagram," in *Proceedings of the Indian Conference on Computer Vision, Graphics and Image Processing*, vol. 22, 2002.

[10] M. Z. Brown, D. Burschka, and G. D. Hager, "Advances in computational stereo," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 25, no. 8, pp. 993–1008, 2003.

[11] J. Matas, O. Chum, M. Urban, and T. Pajdla, "Robust wide-baseline stereo from maximally stable extremal regions," *Image and Vision Computing*, vol. 22, no. 10, pp. 761–767, 2004.

[12] C. Strecha, T. Tuytelaars, and L. van Gool, "Dense matching of multiple wide-baseline views," in *Proceedings of the 9th IEEE International Conference On Computer Vision*, pp. 1194–1201, October 2003.

[13] Q. Chen and G. Medioni, "Volumetric stereo matching method: application to image-based modeling," in *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR '99)*, pp. 1029–1034, Fort Collins, Colo, USA, June 1999.

[14] M. Gong and Y. Yang, "Fast stereo matching using reliability-based dynamic programming and consistency constraints," in *Proceedings of the 9th IEEE International Conference on Computer Vision*, pp. 610–617, October 2003.

[15] M. Lhuillier and L. Quan, "Match propagation for image-based modeling and rendering," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 24, no. 8, pp. 1140–1146, 2002.

[16] H. Wu, Z. Song, J. Yao, L. Li, and Y. Gu, "Stereo matching based on support points propagation," in *Proceeding of IEEE International Conferemce on Information Science and Technology*, pp. 23–25, IEEE, Hubei, China, March 2012.

[17] G. Zeng, S. Paris, L. Quan, and F. Sillion, "Accurate and scalable surface representation and reconstruction from images," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 29, no. 1, pp. 141–158, 2007.

[18] G. Zeng, S. Paris, L. Quan, and M. Lhuillier, "Surface reconstruction by propagating 3D stereo data in multiple 2D images," in *Proceedings of the European Conference on Computer Vision*, pp. 163–174, 2004.

[19] J. Cech and R. Sara, "Efficient sampling of disparity space for fast and accurate matching," in *Proceedings of the International Workshop on Benchmarking Automated Calibration, Orientation, and Surface Reconstruction from Images*, 2007.

[20] L. Wang, H. Jin, and R. Yang, "Search space reduction for MRF stereo," in *Proceedings of the European Conference on Computer Vision*, 2008.

[21] L. Wang and R. Yang, "Global stereo matching leveraged by sparse ground control points," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 3033–3040, June 2011.

[22] C. Harris and M. Stephens, "A combined corner and edge detector," in *Proceedings of the 4th Alvey Vision Conference*, pp. 147–151, 1988.

[23] D. G. Lowe, "Distinctive image features from scale-invariant keypoints," *International Journal of Computer Vision*, vol. 60, no. 2, pp. 91–110, 2004.

[24] D. Comaniciu and P. Meer, "Mean shift: a robust approach toward feature space analysis," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 24, no. 5, pp. 603–619, 2002.

[25] A. Geiger, M. Roser, and R. Urtasun, "Urban Scenes Dataset," 2013, http://www.cvlibs.net/datasets/kitti/eval_stereo_flow.php.

[26] S. M. Seitz, B. Curless, J. Diebel, D. Scharstein, and R. Szeliski, "A comparison and evaluation of multi-view stereo reconstruction algorithms," in *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR '06)*, pp. 519–526, June 2006.

[27] M. Weber, M. Humenberger, and W. Kubinger, "A very fast census-based stereo matching implementation on a graphics processing unit," in *Proceedings of the 12th International Conference on Computer Vision Workshops (ICCV '09)*, pp. 786–793, IEEE, October 2009.

[28] A. Bensrhair, P. Miché, and R. Debrie, "Fast and automatic stereo vision matching algorithm based on dynamic programming method," *Pattern Recognition Letters*, vol. 17, no. 5, pp. 457–466, 1996.

[29] S. Birchfield and C. Tomasi, "Depth discontinuities by pixel-to-pixel stereo," *International Journal of Computer Vision*, vol. 35, no. 3, pp. 269–293, 1999.

[30] Y. Ohta and T. Kanade, "Stereo by intra- and inter- scanline search using dynamic programming," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 7, no. 2, pp. 139–154, 1985.

[31] O. Veksler, "Stereo correspondence by dynamic programming on a tree," in *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR '05)*, pp. 384–390, June 2005.

[32] O. D. Faugeras and R. Keriven, "Complete dense stereovision using level set methods," in *Proceedings of the European Conference on Computer Vision*, June 1998.

[33] K. N. Kutulakos and S. M. Seitz, "Theory of shape by space carving," *International Journal of Computer Vision*, vol. 38, no. 3, pp. 199–218, 2000.

[34] L. Alvarez, R. Deriche, J. Sánchez, and J. Weickert, "Dense disparity map estimation respecting image discontinuities: a PDE and scale-space based approach," *Journal of Visual Communication and Image Representation*, vol. 13, no. 1-2, pp. 3–21, 2002.

[35] C. Strecha, R. Fransens, and L. van Gool, "Combined depth and outlier estimation in multi-view stereo," in *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pp. 2394–2401, June 2006.

[36] S. M. Seitz and C. R. Dyer, "Photorealistic scene reconstruction by voxel coloring," in *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pp. 1067–1073, June 1997.

[37] Y. Boykov, O. Veksler, and R. Zabih, "Fast approximate energy minimization via graph cuts," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 23, no. 11, pp. 1222–1239, 2001.

[38] J. Yedidia, W. T. Freeman, and Y. Weiss, "Understanding belief propagation and its generalizations," in *Proceedings of the International Joint Conference on Artificial Intelligence, Distinguished Papers Track*, 2001.

[39] V. Kolmogorov and R. Zabih, "Multi-camera scene reconstruction via graph cuts," in *Proceedings of the European Conference on Computer Vision*, pp. 82–96, 2002.

[40] C. Rhemann, A. Hosni, M. Bleyer, C. Rother, and M. Gelautz, "Fast cost-volume filtering for visual correspondence and beyond," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR '11)*, pp. 3017–3024, June 2011.

[41] P. Pritchett and A. Zisserman, "Wide baseline stereo matching," in *Proceedings of the 6th International Conference on Computer Vision*, pp. 754–760, IEEE, January 1998.

[42] E. Tola, V. Lepetit, and P. Fua, "A fast local descriptor for dense matching," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR '08)*, pp. 1–8, Anchorage, Alaska, USA, June 2008.

[43] T. Tuytelaars and L. V. Gool, "Wide baseline stereo matching based on local, affinely invariant regions," in *Proceedings of the British Machine Vision Conference*, pp. 412–425, 2000.

[44] H. Bay, A. Ess, T. Tuytelaars, and L. van Gool, "Speeded-Up Robust Features (SURF)," *Computer Vision and Image Understanding*, vol. 110, no. 3, pp. 346–359, 2008.

[45] K. Mikolajczyk and C. Schmid, "A performance evaluation of local descriptors," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 27, no. 10, pp. 1615–1630, 2005.

[46] S. M. Smith and J. M. Brady, "SUSAN: a new approach to low level image processing," *International Journal of Computer Vision*, vol. 23, no. 1, pp. 45–78, 1997.

[47] E. Tola, V. Lepetit, and P. Fua, "DAISY: an efficient dense descriptor applied to wide-baseline stereo," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 32, no. 5, pp. 815–830, 2010.

[48] R. M. Haralick and L. G. Shapiro, "Image segmentation techniques.," *Computer Vision, Graphics, & Image Processing*, vol. 29, no. 1, pp. 100–132, 1985.

[49] G. Otto and T. Chau, "'Region-growing' algorithm for matching of terrain images," *Image and Vision Computing*, vol. 7, no. 2, pp. 83–94, 1989.

[50] M. O'Neill and M. Denos, "Practical approach to the stereo matching of urban imagery," *Image and Vision Computing*, vol. 10, no. 2, pp. 89–98, 1992.

[51] T. Kim and J. Muller, "Automated urban area building extraction from high resolution stereo imagery," *Image and Vision Computing*, vol. 14, no. 2, pp. 115–130, 1996.

[52] M. Lhuillier and L. Quan, "A quasi-dense approach to surface reconstruction from uncalibrated images," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 27, no. 3, pp. 418–433, 2005.

[53] J. Kannala and S. S. Brandt, "Quasi-dense wide baseline matching using match propagation," in *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR '07)*, June 2007.

[54] Z. Megyesi, G. Kós, and D. Chetverikov, "Dense 3D reconstruction from images by normal aided matching," *Machine Graphics and Vision*, vol. 15, no. 1, pp. 3–28, 2006.

[55] J. Čech, J. Sanchez-Riera, and R. Horaud, "Scene flow estimation by growing correspondence seeds," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR '11)*, pp. 3129–3136, June 2011.

[56] C. Vogel, S. Roth, and K. Schindler, "Piecewise rigid scene flow," in *Proceedings of the International Conference on Computer Vision*, 2013.

[57] K. Yamaguchi, D. McAllester, and R. Urtasun, "Robust monocular epipolar flow estimation," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2013.

[58] M. J. Atallah, "Faster image template matching in the sum of the absolute value of differences measure," *IEEE Transactions on Image Processing*, vol. 10, no. 4, pp. 659–663, 2001.

[59] K. Mikolajczyk, T. Tuytelaars, C. Schmid et al., "A comparison of affine region detectors," *International Journal of Computer Vision*, vol. 65, no. 1-2, pp. 43–72, 2005.

[60] J. S. Beis and D. G. Lowe, "Shape indexing using approximate nearest-neighbour search in high-dimensional spaces," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 1000–1006, June 1997.

[61] "Edge Detection and Image Segmentation (EDISON) System," 2014, http://coehttp://www.rutgers.edu/riul/research/code/EDISON/doc/overview.html.

[62] J. Cech and R. Sara, Cech GCS Dataset, 2013, http://cmp.felk.cvut.cz/~cechj/GCS/.

[63] D. Scharstein and R. Szeliski, "Middlebury Stereo Matching Benchmark," 2013, http://vision.middlebury.edu/stereo/.

[64] Q. Yang, L. Wang, and N. Ahuja, "A constant-space belief propagation algorithm for stereo matching," in *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR '10)*, pp. 1458–1465, San Francisco, Calif, USA, June 2010.

Hindawi Publishing Corporation
Mathematical Problems in Engineering
Volume 2014, Article ID 426234, 17 pages
http://dx.doi.org/10.1155/2014/426234

Research Article

Hybrid Particle and Kalman Filtering for Pupil Tracking in Active IR Illumination Gaze Tracking System

Jian-nan Chi,[1,2] Li-hua Xie,[3] Peng-yun Zhang,[1] Yi-fang Lu,[1] and Guo-sheng Zhang[2]

[1] *School of Automation and Electrical Engineering, University of Science and Technology Beijing, Beijing 100083, China*
[2] *Key Laboratory of Operation Safety Technology on Transport Vehicles, Ministry of Transport, Beijing 100088, China*
[3] *School of Electrical and Electronic Engineering, Nanyang Technological University, Singapore 639798*

Correspondence should be addressed to Jian-nan Chi; ustbjnc@ustb.edu.cn

Received 19 February 2014; Revised 26 June 2014; Accepted 1 July 2014; Published 30 September 2014

Academic Editor: Yi Chen

A novel pupil tracking method is proposed by combining particle filtering and Kalman filtering for the fast and accurate detection of pupil target in an active infrared source gaze tracking system. Firstly, we utilize particle filtering to track pupil in synthesis triple-channel color map (STCCM) for the fast detection and develop a comprehensive pupil motion model to conduct and analyze the randomness of pupil motion. Moreover, we built a pupil observational model based on the similarity measurement with generated histogram to improve the credibility of particle weights. Particle filtering can detect pupil region in adjacent frames rapidly. Secondly, we adopted Kalman filtering to estimate the pupil parameters more precisely. The state transitional equation of the Kalman filtering is determined by the particle filtering estimation, and the observation of the Kalman filtering is dependent on the detected pupil parameters in the corresponding region of difference images estimated by particle filtering. Tracking results of Kalman filtering are the final pupil target parameters. Experimental results demonstrated the effectiveness and feasibility of this method.

1. Introduction

Gaze tracking is the technology to get gaze direction or gaze point on the computer screen through mechanical, electronic, optical, and other methods, which can be classified into two different types called the intrusive and the nonintrusive. Gaze tracking is widely used in various applications [1, 2] such as "human computer interaction for disabled people," "virtual reality," "vehicle driver assistance," "human behavior study." Recently, gaze tracking based on the analysis of digital video (video oculography (VOG)) is becoming a popular research topic.

In VOG system, computer vision is used to capture human face images and detect eye features. From these features, the gaze parameters can be extracted to acquire the gaze direction or gaze points. When human eyeball rotates, namely, gaze direction changes, some eye features such as the corner of the eye remain unchanged. Some other features such as pupil center, however, will change correspondingly. In this case, gaze parameters are produced between the changed features and unchanged features, which are used for describing gaze change. The VOG gaze tracking technology has aroused increasing interests of professionals in research and development due to its weak interference to people, simple operation, and high accuracy.

VOG gaze tracking commonly uses pupil center cornea reflection (PCCR) technique based on the active IR illumination. The PCCR method applies an infrared light source to produce cornea reflection (Purkinje spot) and calculates the vector from cornea reflection to the pupil center in captured images. Therefore, the PCCR method can estimate gaze direction through eye structure model or mapping model [3–6]. Thus, VOG gaze tracking is composed of two components: gaze feature parameter extraction and gaze direction estimation.

Eye feature detection in PCCR within VOG system consists of two processes: pupil segmentation and Purkinje location in the neighborhood of pupil. In the existing references, the method of differentiating of bright pupil and dark pupil images is utilized widely in the active illuminator (light

(a) Dark pupil (b) Bright pupil (c) Difference pupil

Figure 1: Bright pupil image, dark pupil image, and difference pupil image.

source) system which is made up of two concentric rings of IR LEDS whose center coincides with the camera optical axis [3–5]. The inner ring LED light source is close to the camera optical axis for producing bright pupil image, which is just like "red eye" in some photographs. The LED light source of outer ring is used to produce Purkinje spot on the surface of the user's eyeball. The inner ring and outer ring LED light alternately to produce bright pupil image and dark pupil image in adjacent frames, which lead to the background elimination via differentiating bright pupil and dark pupil images of adjacent even and odd frames. As a result, pupil is more apparent and easy to be found in the whole face image [6–8]. In differentiating images of bright pupil and dark pupil, threshold method is usually used to segment pupil in local eye region [9–11]. Images of dark pupil, bright pupil, and difference pupil are shown in Figures 1(a), 1(b), and 1(c), respectively.

The extraction of gaze parameters within VOG system mostly relies on pupil location, which is accomplished in two adjacent frames of bright and dark pupils. Operational steps for locating the pupil must be repeated in every two bright pupil and dark pupil frames of video sequence to ensure the accurate gaze parameters. But there will be some drawbacks, shown below, if we choose to detect pupil by scanning each whole image.

(1) A large amount of required calculation to locate pupil results in the worse real time performance of gaze direction estimation.

(2) Since the historical information of pupil motion cannot be used to eliminate the influences of some factors such as eye blinking and external light, it is significantly difficult to extract gaze parameters robustly and get gaze direction estimation accurately.

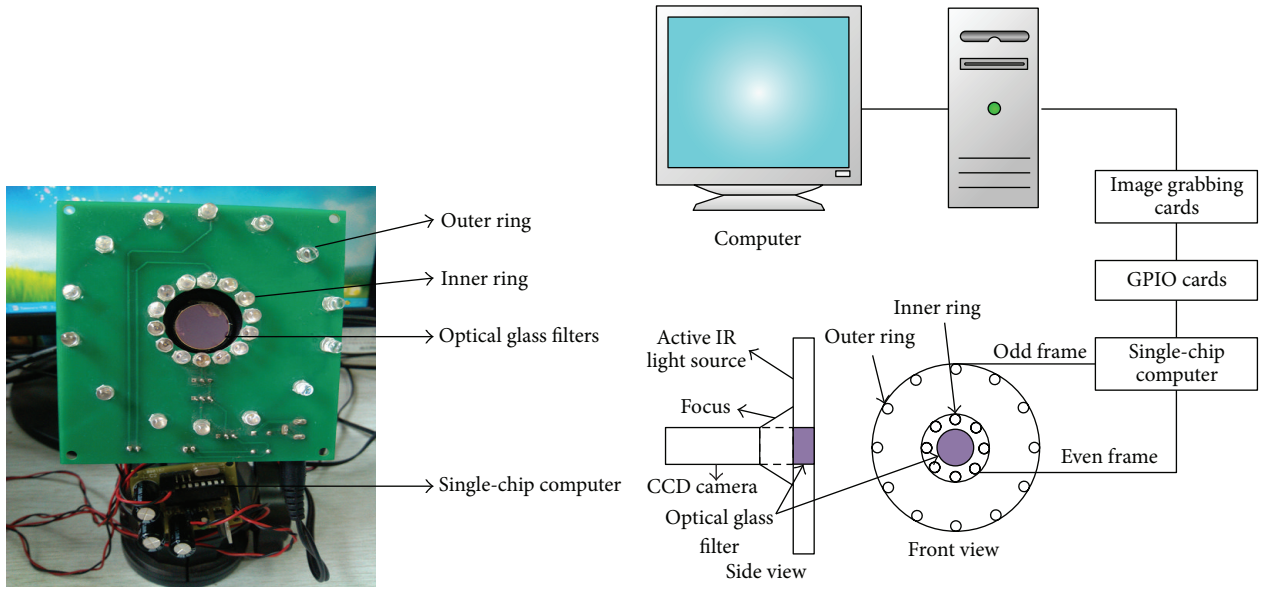
Under some special conditions, such as in the case of eye blinking or eye slightly closed, it is hard to obtain precise pupil parameters by pupil segmentation directly without using the historical information of pupil motion. In order to solve the above two issues, after locating pupil in the initial frame, pupil tracking should be carried out in the following video sequence. Based on pupil tracking results, we can locate and accurately segment pupil to extract pupil parameters. In video sequences, target tracking can be categorized into two different types.

(1) Tracking before detection [12–14]; that is, determine the location of target through tracking algorithm and then finely segment and detect target in the tracking region.

(2) Tracking after detection [15]; that is, estimate the detected target parameters through tracking algorithm using the historical information of target, which can get more precisely stable target parameters.

Currently, most pupil target tracking algorithms in the gaze tracking system implemented in two adjacent frames focus on the pupil fast detection to determine the location of pupil. This way, which can be called tracking before detection, would result in avoiding detecting pupil in the whole image and improving real-time pupil detection. Pupil tracking can be completed and achieved by various proposed techniques such as Kalman filtering, mean shift, and combination of Kalman filtering and "mean shift" [11, 16, 17]. In reality, the fact is that pupil occurs and moves randomly and often disappears especially in some special conditions like eye blinking, so it is complicated and difficult to build an accurate pupil motion model. Both Kalman filtering and "mean shift" have their own defects to deal with pupil tracking especially under some special condition that we discussed above.

Particle filtering is ideal to predict the optimal estimation of a nonlinear and non-Gaussian dynamical system. It can be used to solve the problems of the pupil occlusion and disappearance which are caused by randomness of the eye movement, blinking, and eye closure. For example, Hansen and Pece [18, 19] used particle filtering for iris and pupil tracking. References [20–22] used particle filtering for pupil tracking as well. However, the particle filtering methods above all aim at solving the pupil fast detection, which do not take the pupil shape model and the characteristic of the movement into account. For example, (1) since the shape of pupil is ellipse, the foreground and background cannot be distinguished with a rectangular model; (2) they never consider the geometric similarity of the pupil target ellipse as important clue in the research.



(a) Real gaze tracking system

(b) Sketch map of gaze tracking system

Figure 2: Gaze tracking system.

Real-time pupil tracking is the common problem in the VOG system. The considerations for effective tracking are listed as follows.

(1) Since the size of pupil is very small and often influenced by eyelashes and eyelids, one of the most important functions of pupil target model in tracking process is to minimize background interference and distinguish target and background.

(2) Because of the random pupil motion which is caused by head movement and pupil rotation, transition equation must always attempt to achieve the detailed characteristics of pupil motion as far as possible.

In this paper, a costless and widely used gaze tracking system using PCCR technique is developed for the research. To achieve fast and precise pupil detection, a pupil tracking method based on hybrid of particle filtering and Kalman filtering (HPFKF) in the VOG system is proposed and qualified to meet the requirements as mentioned. The main work in this paper includes the following.

(1) Particle filtering is used to track pupil for the fast determination of the accurate pupil location.

(2) Kalman filtering is utilized to estimate pupil parameters for pursuing the precision in parameter detection integrated with particle filtering location, which could improve the accuracy and reliability of pupil parameters.

2. Gaze Tracking System and the Initial Pupil Detection Method

In this paper, the gaze tracking system is composed of an infrared light source with two concentric rings, optical glass filters, CCD camera, and image grabbing cards, GPIO (general purpose input/output) cards, single-chip computer, computer, and display screen. There is an inner ring within the infrared light source that produces a bright pupil image when the inner ring is turned on. Alternately, when the outer ring is on, it produces a dark pupil and Purkinje image. When the user watches the screen, the CCD camera obtains face images and sends them to the PC, in which eye features are extracted and eye focus is gotten by gaze mapping function while gaze points are displayed on the screen. To capture the face images that enable this process, the GPIO card is used to obtain the frame alignment signal of the CCD video sequence and employs a single chip microcomputer to control the switch of the light source, activating the inner and outer LED rings alternatively. Gaze tracking system is shown in Figure 2.

To estimate the gaze points on the screen, the first step is to detect gaze parameters in image, in which pupil detection plays an important role. In the initial two frames which CCD camera grabbed, the pupil is located firstly and its characteristic parameters are detected. In the following video image sequences pupil tracking is carried out. The scheme of initial pupil detection and characteristic parameters extraction are described as in Figure 3.

3. The Pupil Tracking Based on Combination of Particle Filtering and Kalman Filtering

3.1. Framework of HPFKF. The procedure of parameters extraction proposed in Section 2 is completed in the first two adjacent frames. We capture the pupil in the two initial frames firstly and then perform pupil tracking in the following video sequences.

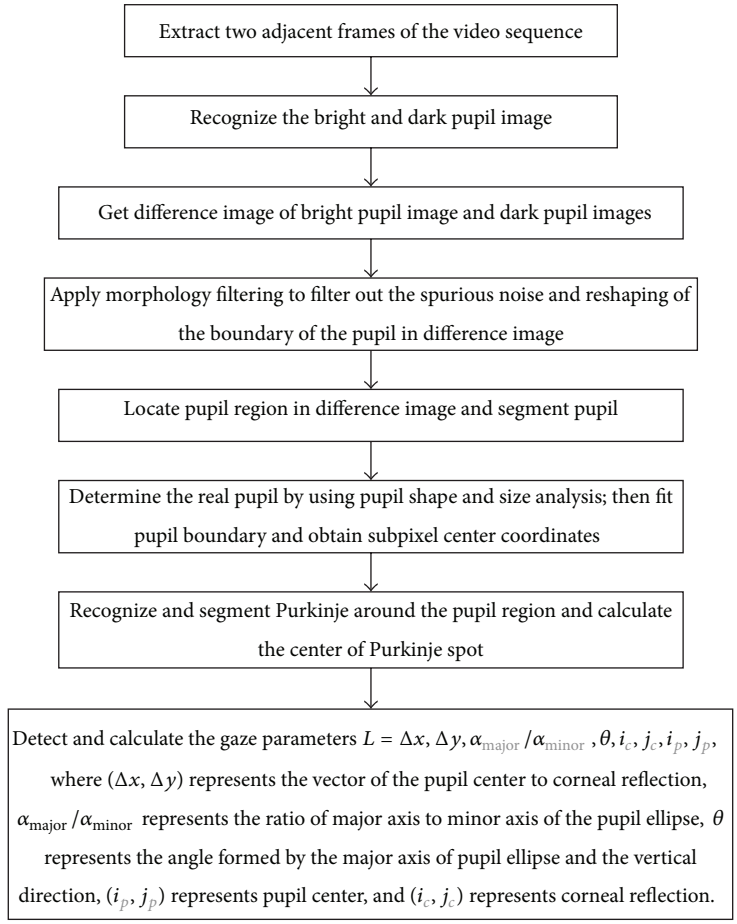


Figure 3: Sketch map of initial pupil detection.

In this paper it is shown that pupil target estimation is accomplished by the combination of particle filtering [19] and Kalman filtering by using of bright pupil image, dark pupil image and difference image in adjacent frames. The latter process tracks for accurate estimation of pupil parameters, whereas particle filtering tracks for fast location of pupil. The objective of this proposed approach is to improve real-time performance of pupil detection by particle filtering and improve accuracy of pupil parameters detection by Kalman filtering, which is carried out in every two adjacent frames.

The flow chart of pupil target tracking method based on the combination of particle filtering and Kalman filtering is shown in Figure 4.

During this tracking process, four adjacent frames images (bright pupil, dark pupil, bright pupil, and dark pupil, resp.) are generated by system hardware and defined as follows: the first two frames (bright pupil and dark pupil) are called previous image frame, and the following two frames are called next image frame. Difference image of previous image frame is the subtraction of the bright pupil image and the dark pupil image of the previous image. Difference image of next image frame is the subtraction of the bright pupil image and the dark pupil image of the next image frame. In the initial two frames of bright pupil and dark pupil image, the following steps are implemented: firstly, find the difference of the bright and dark pupil images; secondly, segment pupil in difference image by use of the pupil detection method introduced in Section 2; thirdly, detect pupil target parameters; and finally, accomplish the tracking initialization. Hence, according to the definition of previous image frame and next image frame, the pupil tracking process is implemented in these resulting images.

In previous and next image frames, triple-channel color image is synthesized by bright pupil, dark pupil, and the corresponding difference images. The pupil tracking based on particle filtering is achieved in synthesized color image made from previous image frame and next image frame. According to the detected pupil parameters in previous image frame, we can predict particle state in triple-channel color image synthesized by next image frame. The process of particle filtering pupil tracking can determine the location of pupil. The tracking result based on particle filtering plays a guiding role for the Kalman filtering pupil parameters estimation which is accomplished in the difference image made from previous image frame and next image frames. The Kalman state transition equation of pupil motion is

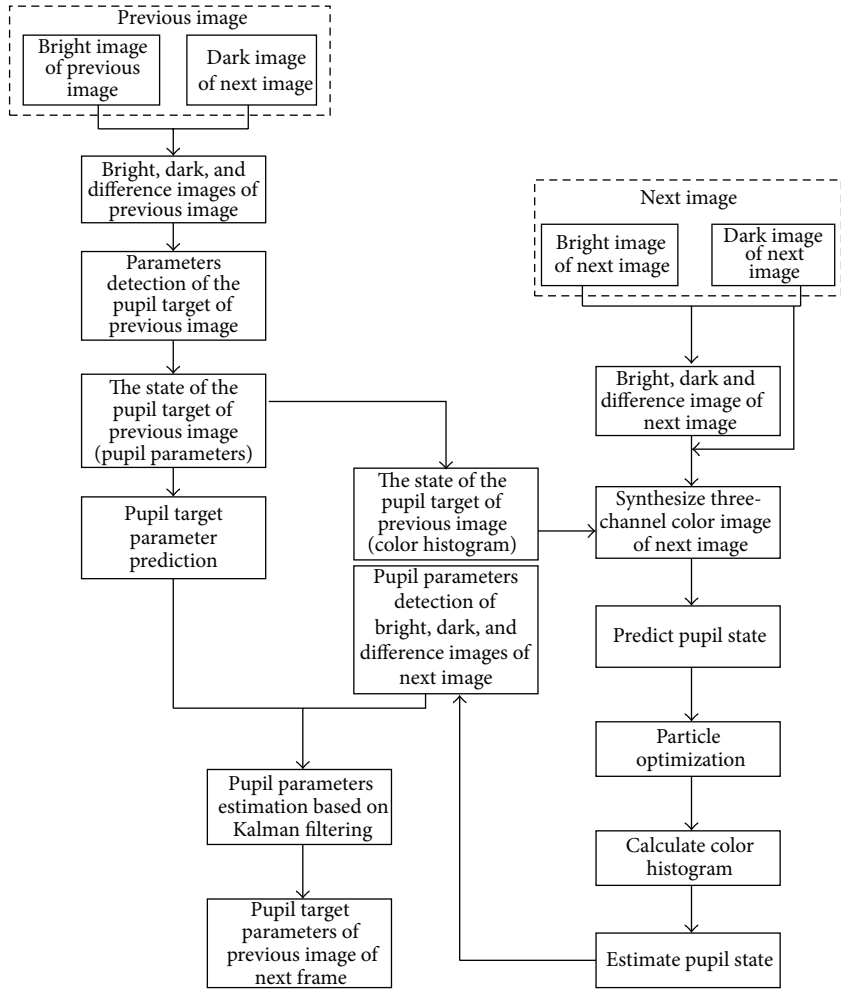


Figure 4: The flow chart of pupil tracking method based on a combination of particle filtering and Kalman filtering.

determined by particle filtering tracking result, which can predict pupil target parameters. According to particle filtering tracking results, pupil can be segmented and pupil parameters can be extracted to serve as observation for the correction of the Kalman prediction above. Finally, the Kalman filtering process can estimate the pupil parameters of next difference image. Tracking result of Kalman filtering is not only the results of final pupil detection, but also the target parameters in the next previous image frame.

3.2. Review of Kalman Filtering and Particle Filtering

(1) Particle Filtering. Particle filtering [23, 24] is an approximation algorithm based on Bayesian estimation of sampling theory, which combines sequential Monte Carlo (SMC) method together with the Bayesian theory.

Let the state parameter vector of a target at time t be denoted by x_t and its observation by z_t. The history of observation from time 1 to t is denoted by $Z_t = [z_1, z_2, \ldots, z_t]$. The Bayesian formulation of particle filtering is expressed as

$$p(x_t \mid Z_t) \propto p(z_t \mid x_t) \cdot p(x_t \mid Z_{t-1}) = p(z_t \mid x_t) \cdot \int p(x_t \mid x_{t-1}) \cdot p(x_{t-1} \mid z_{t-1})\, dx_{t-1}. \quad (1)$$

The basic idea of particle filter is to find a set of random samples in the state space of the posterior probability density $p(x_t \mid Z_t)$ approximation and to replace $E[g(x_t \mid z_t)]$ with the sample mean to obtain a state of minimum variance estimate.

The key point of implementation of particle filter is to find random samples of obeying the distribution of $p(x_t \mid z_t)$, which are called particles. N sampling points are extracted from the posterior probability density independently, which represented the posterior probability density by the weighted sum.

Bayesian importance sampling (BIS) is to use an easy sample of known distribution $q(x_t \mid z_t)$ to replace posterior probability density for sampling by weighing the sampling

particles of the importance function to approximate $p(x_t \mid z_t)$. From the Bayes theory we obtain

$$\begin{aligned} E(g(x_t)) &= \int g(x_t) \frac{p(z_t \mid x_t)\, p(x_t)}{q(x_t \mid z_t)\, p(z_t)} q(x_t \mid z_t)\, dx_t \\ &= \int g(x_t) \frac{w_k(x_t)}{p(z_t)} q(x_t \mid z_t)\, dx_t \\ &= \frac{\int (g(x_t)\, w_k(x_t))\, q(x_t \mid z_t)\, dx_t}{\int w_k(x_t)\, q(x_t \mid z_t)\, dx_t}. \end{aligned} \tag{2}$$

After sampling from the importance function, mathematical expectation of the target state vector is approximated as

$$\begin{aligned} \overline{E(g(x_t))} &= \frac{(1/N) \sum_{i=1}^{N} g\left(x_t^{(i)}\right) w_k\left(x_t^{(i)}\right)}{(1/N) \sum_{i=1}^{N} w_k\left(x_t^{(i)}\right)} \\ &= \sum_{i=1}^{N} g\left(x_t^i\right) \widetilde{w}_k\left(x_t^i\right), \end{aligned} \tag{3}$$

where $\widetilde{w}_k(x_t^i) = w_k(x_t^{(i)}) / \sum_{i=1}^{N} w_k(x_t^{(i)})$ is normalized weights and $x_t^{(i)}$ is a sampling particle from $q(x_t \mid z_t)$. To overcome the deficiencies of importance sampling, resampling technique is often used for sampling. The basic idea of resampling is to suppress or eliminate small weight particles. Big weight particles are replicated according to their weights. With the dynamic/temporal propagation, particle filtering has been widely used for tracking applications.

(2) Kalman Filtering. Kalman filtering [25, 26] is based on state space model and state space equations of linear dynamical systems, providing a recursive solution of linear optimization filtering. Using the estimated value of the previous time step and the observed value of the current time step to update the target state estimation, we can get the estimated target state value of the current time.

Kalman filter is an optimal linear recursive filter according to minimum mean square error rule. Its dynamic system is described by a state equation and observation equation. The state equation is

$$X_k = \Phi_{k,k-1} X_{k-1} + W. \tag{4}$$

The observation equation is

$$Z_k = H_k X_k + V, \tag{5}$$

where X_k is the state vector in time k, Y_k is the observation vector in time k, $\Phi_{k,k-1}$ is the state transition matrix from time $k-1$ to time k, H_k is the measurement matrix in time k, W is state noise vector, and V is observation noise vector. Assume that the process noise covariance is Q, the observation noise covariance is R. Combining the predicted value with measured value of the state, the estimated state value of the current time can be obtained:

$$\begin{aligned} X_k &= X_{k,k-1} + Kg_k \left(Z_k - H_k X_{k,k-1}\right) \\ P_{k,k} &= \left(I - Kg_k H\right) P_{k,k-1}, \end{aligned} \tag{6}$$

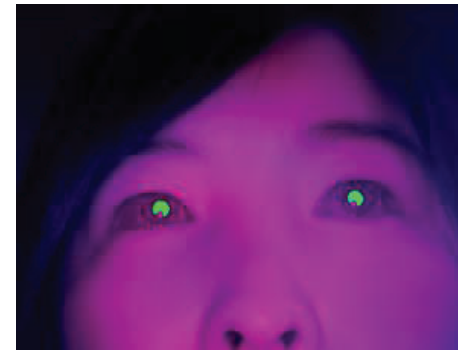


Figure 5: Synthesis triple-channel color map.

where $Kg_k = P_{k,k-1}H' / (HP_{k,k-1}H' + R)$ is Kalman gain and $P_{k,k-1} = \Phi P_{k-1,k-1} \Phi' + Q$ is corresponding covariance to $X_{k,k-1}$. The above algorithms constitute the recursive structure of Kalman filtering.

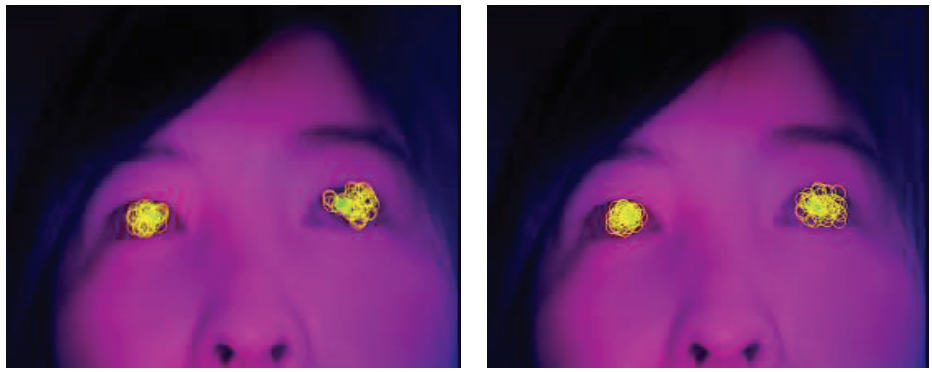
3.3. Pupil Tracking Based on Particle Filtering

3.3.1. Triple-Channel (HSV) Color Map. In order to make the pupil region more apparent in the tracking process, we propose the employment of a synthesized triple-channel color map (STCCM) based on bright pupil image, dark pupil image, and corresponding difference images in which particle filtering pupil tracking can be achieved. STCCM is defined as follows: (1) chrominance of STCCM is equivalent to that of difference image in order to highlight the pupil region color; (2) saturation of dark pupil image is representative of saturation of STCCM; and (3) STCCM brightness is equal to the average brightness of bright pupil and dark pupil images to confirm that the brightness of STCCM is between that of bright pupil and dark pupil images, which prevents significant change in brightness caused by the external light source. Synthesized triple-channel color map is shown in Figure 5. We know that tracking in bright pupil or dark pupil image is difficult, where the pupil feature is not obvious (pupil gray is close to the rest of face). However, STCCM takes full advantage of the triple-channel information, making the color feature of pupil apparent and significantly different from the rest of face. The purpose of the proposed STCCM is to improve the accuracy and stability of pupil tracking.

3.3.2. Pupil Target Model. Pupil target model is established for particle filtering tracking; namely the state of pupil target is described by the vector $\mathbf{X}_t$ as follows:

$$\mathbf{X}_t = \left(c_x, c_y\right), \tag{7}$$

where c_x, c_y represent image coordinates of the center of the pupil ellipse.



(a) STATES_{t-1} (b) STATES_t

Figure 6: STATES_{t-1} and STATES_t.

3.3.3. Particle Initialization. The initial state of particle [27] is

$$\text{STATES}_{t-1} = \left\{ \left(\text{state}_{t-1}^n, \pi_{t-1}^n\right) \mid \text{state}_{t-1}^n = X_{t-1} + V_{t-1}, \ \pi_{t-1}^n = \frac{1}{N} \right\}, \tag{8}$$

where STATES_{t-1} represents the sample set, $(\text{state}_{t-1}^n, \pi_{t-1}^n)$ is the state and weight of each particle of the set, respectively, and V_{t-1} is noise vector. STATES_{t-1}, that is, state of time $t-1$, is shown in Figure 6(a).

3.3.4. Particle Selection. According to the weight distribution, randomly select N status from STATES_{t-1}. During filtering, the particles with higher weight have a greater chance of being selected and are chosen more often. Hence, the particles with low weight are eliminated during the selection process:

$$\widetilde{\text{STATES}}_{t-1} = \left\{ \left(\widetilde{\text{state}_{t-1}^n}, \phi\right) \mid \widetilde{\text{state}_{t-1}^n} \sim \text{STATES}_{t-1} \right\}. \tag{9}$$

3.3.5. Pupil Motion Model. Taking into account the changes of pupil state in a variety of situations, it is possible to establish a simple linear model to describe pupil motion. The law of pupil state movement state_t^n is as follows: the elliptical center position of pupil at time t is representative of the sum of the elliptical center position of pupil at time $t-1$ and displacement of the pupil movement. In addition, when the state of particles is predicted, noise vector should be considered. Then we can obtain the equation of the pupil motion:

$$\text{STATES}_t = \left\{ \left(\text{state}_t^n, \phi\right) \mid \text{state}_t^n = A\,\widetilde{\text{state}_{t-1}^n} + V_{t-1}^n \right\}. \tag{10}$$

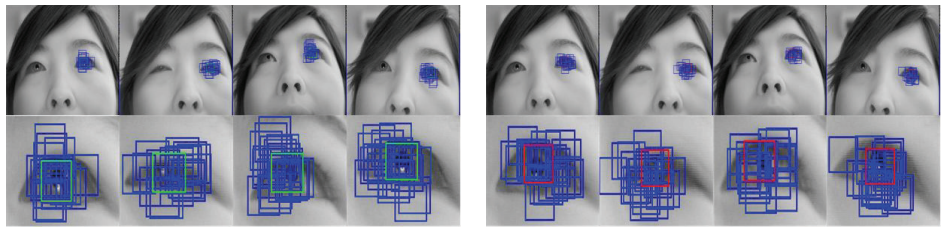
By predicting the state of particles according to pupil motion equation, we can get the particle collection STATES_t at current moment t as shown in Figure 6(b).

3.3.6. Particle Optimization. After iterations for many times, the weights of many particles are smaller, which means that the weight variance of the particle is bigger. The target tracking may be inaccurate when the target is estimated by these particles with very small weights. Furthermore, we can see the phenomenon of particle degeneration. Currently, there are three methods to solve this problem: (1) choose the appropriate sampling density function; (2) use the important resample method [28]; and (3) increase the particle number. Reference [20] has proved that the best sampling density function is the posterior probability density function, which is not easy to sample. Generally the sampling density function is represented by the prior probability density which is notably different from the real probability density. The important resampling method that Gordon et al. [29] have proposed can be described as follows: the particles with high weight reproduce the new particle many times, while the smaller weighted particles reproduce fewer particles. As a result, particle variety is reduced. Although the important resampling method can avoid the particle degeneration, it may cause sampling impoverishment. In addition, the method of increasing particle number can increase the computational burden and influence the real-time tracking.

In this paper, we use the Camshift [30] algorithm to deal with the problem of particle degeneration. The Camshift algorithm (continuously adaptive mean shift) is a nonparameter iteration technology based on a probability distribution model and uses color histogram to represent the object's observation model. The purpose of the Camshift method used in particle filtering is to accomplish particle aggregation. In other words, iterated particles will move to the region of the target, which reduce the amount of particles with smaller weight. Even if the particle number is small, it can reflect the target observation model. An experiment in which 20 particles are used to track pupil in 600 image frames is applied to verify the effectiveness of the Camshift based particle optimization method.

RMSE (root of mean square error) of tracking results shows that the Camshift based particle optimization method can improve the tracking performance.

Figure 7 has shown the tracking results of frames 15, 177, 310, and 544. Table 1 is the experimental results of RMSE, which is the average coordinates of particle states.



(a) The particle filtering method (b) The proposed method

Figure 7: The tracking results.

Table 1: The experiment results of RMSE.

Method	15 frames (pixel)	177 frames (pixel)	310 frames (pixel)	544 frames (pixel)
RMSE (PF method)	256.031	221.181	189.594	305.321
RMSE (the proposed method)	250.008	213.677	184.000	289.249

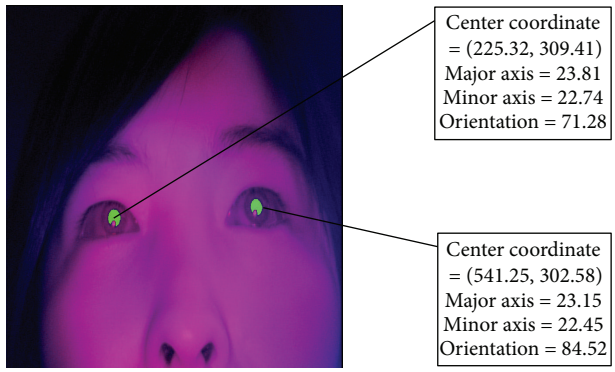


Figure 8: The initial state of pupil.

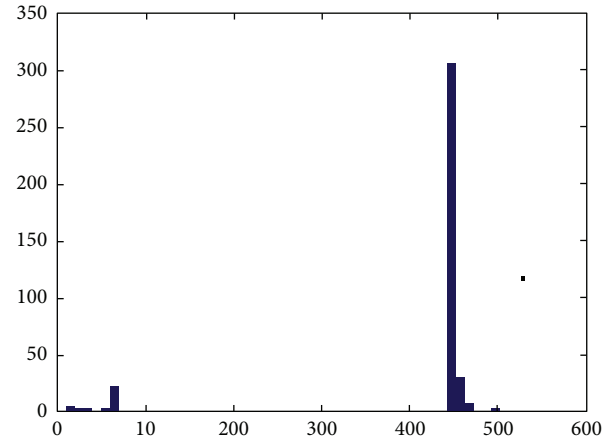


Figure 9: Pupil histogram.

3.3.7. Pupil Observation Model. Color distribution is suitable to be used as target model since it is robust and maintains the persistence of nonrigid target deformation, rotation, and scale changes. Thus, we consider the color weighted histogram of pupil region as our observation model. Epanechnikov kernel function highlights the contributions of different pixel to color histogram. The greatest contribution to the color histogram comes from the existence of pixels located at target center, while the smallest contribution occurs at edge region which may contain background.

Here we use $d(x) = ((x_x - c_x)/a_{\text{major}})^2 + ((x_y - c_y)/a_{\text{minor}})^2$ as distance measurement [31], which is known as "gray contour curve distance" and can measure the important weight of the pupil pixel. This means that the smaller the $d(x)$ is, the greater the pixel weights are. The ellipse curve is composed of pixels of the same $d(x)$. That is to say, the points of this curve have the same gray value. The $d(x)$ of pupil ellipse contour is 1, which gradually decreases towards the pupil center until $d(x)$ is equal to 0, where the "gray contour curve" changes to the center of pupil ellipse. The initial state of pupil and pupil histogram are shown in Figures 8 and 9.

During the update procession, we calculate the weighted histogram for each particle and discover that when there are similar histograms, there is higher weight. The weight of each particle π_t^n can be calculated based on the weighted histogram $H_{\text{state}_t^n}$ of each particle [32]:

$$\pi_t^n = \frac{1}{\sqrt{2\pi}\sigma} e^{(-(1-\rho(H_{\text{state}_t^n}, H_{X_{t-1}}))/2\sigma^2)}. \tag{11}$$

3.3.8. Pupil State Estimation. Pupil target state is estimated by

$$E(X_t) = \frac{\sum_{n=1}^{N} \pi_t^n \text{state}_t^n}{\sum_{n=1}^{N} \pi_t^n}. \tag{12}$$

Firstly, the histogram $H_{E(X_t)}$ of the estimated target is calculated. Then, we get the similarity $\rho(H_{E(X_t)}, H_{X_{t-1}})$, and finally, we obtain the probability weight π_t^E. If probability weight π_t^E of the estimated target is larger than threshold π_T, the tracking is considered to be successful, and target state is updated by ratio α. Contrarily, if the tracking is deemed a failure, then the target state cannot be updated. The tracking result is shown in Figure 10.

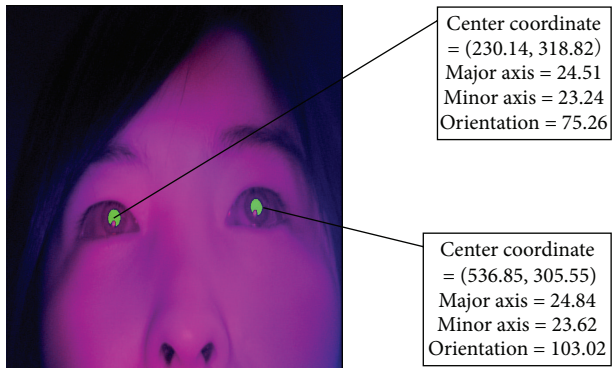


Figure 10: Tracking result.

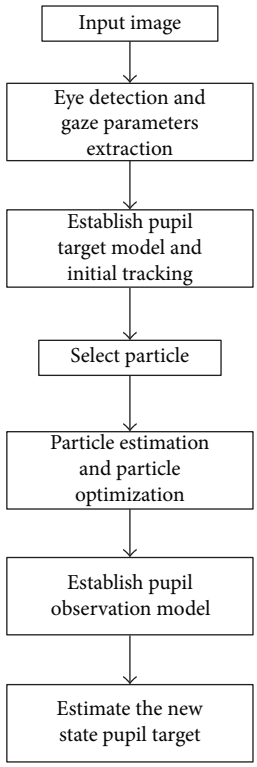


Figure 11: The flow chart of particle filtering tracking.

3.3.9. The Estimation Process of Particle Filtering. The flow chart of the pupil tracking based on particle filtering is shown in Figure 11.

(1) Synthesize triple-channel color image.

(2) Establish pupil model:

$$X_t = \left(c_x, c_y\right). \tag{13}$$

(3) Initialize the state of each particle of the set

$$\mathrm{STATES}_{t-1} = \left\{ \left(\mathrm{state}_{t-1}^n, \pi_{t-1}^n\right) \mid \mathrm{state}_{t-1}^n = X_{t-1} + V_{t-1}, \; \pi_{t-1}^n = \frac{1}{N} \right\}. \tag{14}$$

(4) Select particle.

According to the weight distribution, select the higher weight particles from STATES_{t-1}; then constitute $\widetilde{\mathrm{STATES}}_{t-1}$:

$$\widetilde{\mathrm{STATES}}_{t-1} = \left\{ \left(\widetilde{\mathrm{state}_{t-1}^n}, \phi\right) \mid \widetilde{\mathrm{state}_{t-1}^n} \sim \mathrm{STATES}_{t-1} \right\}. \tag{15}$$

(5) Build pupil motion model.

According to pupil motion model, particle state is predicted; it can also describe pupil motion state by using a simple linear model:

$$\mathrm{STATES}_t = \left\{ \left(\mathrm{state}_t^n, \phi\right) \mid \mathrm{state}_t^n = A\widetilde{\mathrm{state}_{t-1}^n} + V_{t-1}^n \right\}. \tag{16}$$

(6) Establish pupil observation model.

Calculate the histogram $H_{\mathrm{state}_t^n}$ of each particle and then calculate weight π_t^n of each particle based on histogram:

$$\pi_t^n = \frac{1}{\sqrt{2\pi}\sigma} e^{\left(-\left(1-\rho\left(H_{\mathrm{state}_t^n}, H_{X_{t-1}}\right)\right)/2\sigma^2\right)}. \tag{17}$$

(7) Estimate pupil state:

$$E\left(X_t\right) = \frac{\sum_{n=1}^{N} \pi_t^n \mathrm{state}_t^n}{\sum_{n=1}^{N} \pi_t^n}. \tag{18}$$

3.4. Pupil Tracking Based on Kalman Filtering. After pupil tracking based on particle filtering, we can determine the location of pupil. As shown in Figure 12, the rectangle describes the location of pupil. Next, we segment the pupil in the corresponding pupil region in difference image and attain pupil parameters to be used as observation of next image frame. Kalman filtering [33] is then used to precisely estimate the pupil parameters based on pupil parameters of the previous frame image and the observation of the next frame image. Hence, Kalman filtering tracking is implemented in the difference images.

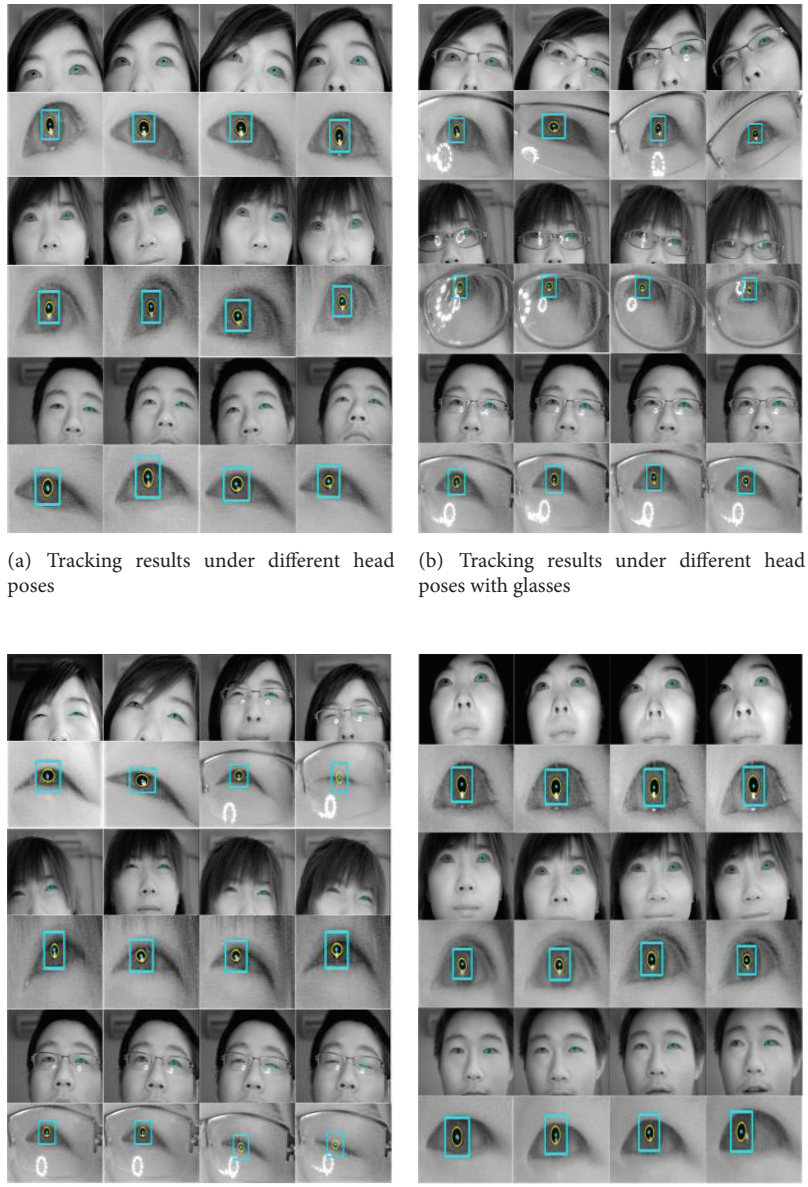
According to the theory of Kalman filtering, we assume that the pupil model in the difference image at time t can be represented as follows:

$$X_t = \left(i_t, j_t, a\mathrm{major}_t, a\mathrm{minor}_t, \theta_t, vx_t, vy_t, \dot{a}_t, \dot{\theta}_t\right), \tag{19}$$

where i_t, j_t are representative of the pupil center coordinates at time t; then $a\mathrm{major}_t, a\mathrm{minor}_t$ are the major axis and minor axis of the pupil ellipse, respectively. θ_t is the angle between the major axis of pupil ellipse and the vertical direction. vx_t, vy_t represent their respective velocity at time t in x, y direction. $\dot{a}_t, \dot{\theta}_t$ signify the rate of change of pupil scale and angle θ_t, respectively. Pupil location estimated by particle filtering at time t is (c_{xt}, c_{yt}). Assuming pupil position parameters estimated in previous image frame being (c_{xt-1}, c_{yt-1}), we can get $vx_t = (c_{xt} - c_{xt-1})/\Delta t$ and $vy_t = (c_{yt} - c_{yt-1})/\Delta t$.

The state vector $\mathbf{X}_t$ at the next time t frame is linearly related to previous state $\mathbf{X}_{t-1}$ by the system motion model as follows:

$$\mathbf{X}_t = \mathbf{A}\mathbf{X}_{t-1} + \mathbf{W}_t. \tag{20}$$



(a) Tracking results under different head poses

(b) Tracking results under different head poses with glasses

(c) Tracking results under blinking

(d) Tracking results under normal condition

Figure 12: Experiment result based on the proposed method.

Here, $\mathbf{A}$ is the state transition matrix, $\mathbf{W}_t$ represents system perturbation which is normally distributed as $P(W_t) \sim N(0, Q)$, and Q represents the process noise covariance.

If we assume that observation vector of pupil target is $\mathbf{Z}_t = (i_t, j_t, a\text{major}_t, a\text{minor}_t, \theta_t)$ which is the pupil parameters detected at time t, we can see that the observation model of the Kalman filtering is

$$\mathbf{Z}_t = \mathbf{H}\mathbf{X}_t + \mathbf{V}_t. \tag{21}$$

Here matrix $\mathbf{H}$ is called "observation matrix," which represents the relationship between current state and current measurement, while $\mathbf{V}_t$ represents measurement uncertainty. $\mathbf{V}_t$ is normally distributed as $p(V_t) \sim N(0, R)$, and R is the measurement noise covariance. Here $\mathbf{H}$ is defined as follows:

$$\mathbf{H} = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \end{bmatrix}. \tag{22}$$

Given the state transition model (see (20)), the observation model (see (21)) and the initial condition, the state vector $\mathbf{X}_t$ and its covariance matrix P_t can be updated as follows.

Let $\widehat{\mathbf{X}}_{t|t-1}$ be the estimated state at time t, resulting from the transition of system state model. $\widehat{\mathbf{X}}_{t|t-1}$ is often referred to as the "prior state estimate." $\widehat{\mathbf{X}}_{t|t}$ is estimated using both the system model (see (20)) and the measurement model (see (21)) and is usually referred to as the "posterior state estimation." Let $\mathbf{P}_{t|t-1}$ and $\mathbf{P}_{t|t}$ be the covariance matrices for the state estimates of $\widehat{\mathbf{X}}_{t|t-1}$ and $\widehat{\mathbf{X}}_{t|t}$, respectively. These matrices determine the uncertainties associated with the prior and posterior state estimates. The Kalman filtering algorithm for state prediction and updating may be summarized below.

3.4.1. State Prediction. According to estimation state $\widehat{\mathbf{X}}_{t-1|t-1}$ and its covariance $\mathbf{P}_{t-1|t-1}$ at $t-1$ time, $\widehat{\mathbf{X}}_{t|t-1}$ and its covariance $\mathbf{P}_{t|t-1}$ at t time can be predicted:

$$\begin{aligned}\widehat{\mathbf{X}}_{t|t-1} &= \mathbf{A}\widehat{\mathbf{X}}_{t-1|t-1},\\ \mathbf{P}_{t|t-1} &= \mathbf{A}\mathbf{P}_{t-1|t-1}\mathbf{A}^T + \mathbf{Q}_t.\end{aligned} \tag{23}$$

Here, $\mathbf{A}$ represents state transition matrix. In Section 3.3 of this paper, we can determine the location of pupil by the method of particle filtering. Then, in the corresponding pupil position in the difference image of next image frame, we segment pupil target into binary image and calculate the circularity of the pupil target:

$$R_t = \frac{4\pi S_t}{L_t^2}. \tag{24}$$

Here, S_t is the area of the pupil target, while L_t is the perimeter. Both of them are described by pixel number. The range of value R_t is $[0, 1]$, being that the rounder the target is, the larger the R_t is, and the R_t value of a standard circle is 1. The shape of the pupil ellipse can be described by circularity.

During pupil motion process, both its shape and scale change. Assuming that R_t is the target circularity of the region that Particle filtering tracked at time t and R_{t-1} at time $t-1$, the ratio of R_t and R_{t-1} would be defined as follows:

$$k = \frac{R_t}{R_{t-1}}. \tag{25}$$

According to different value of k, we can get the different state transition matrix:

$$k > 1: \quad A = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & \Delta t & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & \Delta t & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & -\Delta t & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & \Delta t \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & \Delta t \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix}$$

$$0 < k < 1: \quad A = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & \Delta t & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & \Delta t & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & \Delta t & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & \Delta t \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & \Delta t \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix}. \tag{26}$$

3.4.2. State Updating. Based on particle filtering tracking results (Section 3.3), the prior state estimate $\widehat{\mathbf{X}}_{t|t-1}$ and its covariance matrix $\mathbf{P}_{t|t-1}$, we can determine the region of the pupil, and therefore we segment the pupil and attain pupil parameters $\mathbf{Z}_t$. We can then perform state updating to derive the final state and its covariance matrix. The first task during state updating is to compute the Kalman gain $\mathbf{K}_t$:

$$\mathbf{K}_t = \frac{\mathbf{P}_{t|t-1}\mathbf{H}^T}{\mathbf{H}\mathbf{P}_{t|t-1}\mathbf{H}^T + \mathbf{R}_t}. \tag{27}$$

The gain $\mathbf{K}_t$ can be regarded as a weighing factor when determining the contribution of measurement $\mathbf{Z}_t$ and prediction $\mathbf{H}\widehat{\mathbf{X}}_{t|t-1}$ to the posterior state estimate $\widehat{\mathbf{X}}_{t|t}$. $\widehat{\mathbf{X}}_{t|t}$ is computed as follows:

$$\widehat{\mathbf{X}}_{t|t} = \widehat{\mathbf{X}}_{t|t-1} + \mathbf{K}_t\left(\mathbf{Z}_t - \mathbf{H}\widehat{\mathbf{X}}_{t|t-1}\right). \tag{28}$$

The a posteriori error covariance estimate is computed as follows:

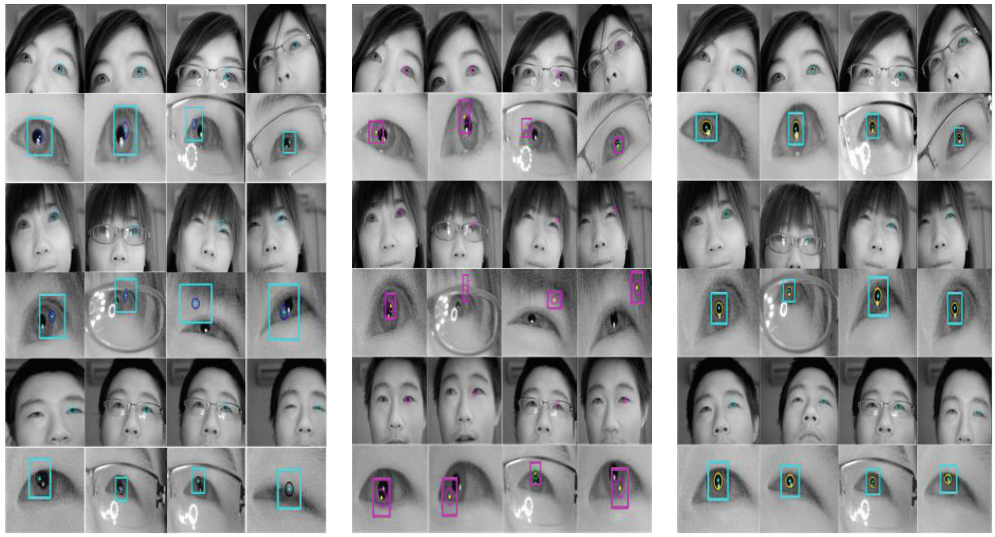
$$\mathbf{P}_{t|t} = \left(\mathbf{I} - \mathbf{K}_t\mathbf{H}\right)\mathbf{P}_{t|t-1}. \tag{29}$$

Kalman filtering always produces complete recursive state estimation through the processes of state prediction and measurement updating. Consequently, the posterior estimation at current moment is a priori estimation of next moment.

4. Experimental Results and Analysis

4.1. Results of the Pupil Tracking. In order to prove the validity of the tracking procedure proposed in this paper, we chose 1000 frames consecutive image sequences offered by 3 users for our experiments. An Intel Pentium Dual, CPU E2200 with 2.20 GHz frequency hardware configuration was used to create image of which resolution is 760 by 576 pixels. The experimental circumstances included indoor, cloudy, and normal fluorescent lamp illumination without strong light interference. 1000 frames tracking results of every user are recorded each time.

During tracking process, the main purpose is to validate the tracking ability of pupil under various conditions. Figure 12(a) shows the tracking results of frames 169, 468, 715, and 856 under significant facial pose changes. Clearly, it is seen that the proposed method can stably track pupil under the conditions of different head poses. Figure 12(b) shows the tracking results of frames 167, 465, 744, and 955



(a) Particle filtering method (b) Direct detection method (c) The proposed method

Figure 13: Tracking results based on three methods.

when there are significant face pose changes with glasses. In these situations, if the light glaring on the glasses does not completely occlude the pupils, our method will effectively track pupil. Figure 12(c) shows the tracking effect of frames 166, 310, 519, 767, and 989 under significant face pose changes when blinking. If there is no long time eye closure, or there is no sudden head movement, the proposed method in gaze tracking system can accurately track pupil and meet real-time requirement. Figure 12(d) shows the tracking effect of frames 82, 454, 684, and 946 under normal conditions. These results demonstrate that this method of combining the two tracking techniques produces much better tracking results than using the direct detection method and particle filtering method individually.

4.2. Analysis of Tracking Performance. To illustrate the differences among particle filtering method, direct detection method, and our proposed method, we show the tracking results of these methods under different conditions. As can be seen from Figure 13(a), particle filtering can be used to track the eye during the whole sequence under different conditions, but it is not always successful in detecting the correct eye position. In Figure 13(a), we also can see that particle filtering can only recognize the center of the rectangle range as pupil center, which leads to the center deviating from the pupil center. As shown in Figure 13(b), the direct detection method cannot correctly detect pupil when one is blinking or wearing glasses. In contrast, the proposed method then shown in Figure 13(c) can accurately obtain the pupil center under different conditions.

Table 2 shows experimental results of RMSE for the detection of pupil center using three methods (particle filtering, direct detection, and the proposed method). In the experiment, we are able to locate the true center of pupil frame by frame manually and determine the real image coordinate of pupil center. From above testing sequences, RMSE between the center of the tracking result and the true center of pupil target can be calculated as shown in Figure 14. In Figure 14, particle filtering and direct detection may be able to track pupil target, but the results are not accurate. In the entire tracking process, the proposed method usually maintains stable and accurate tracking and has a smaller margin of error, which meets the accuracy requirement of the pupil tracking. Figure 15 shows the final gaze point estimation

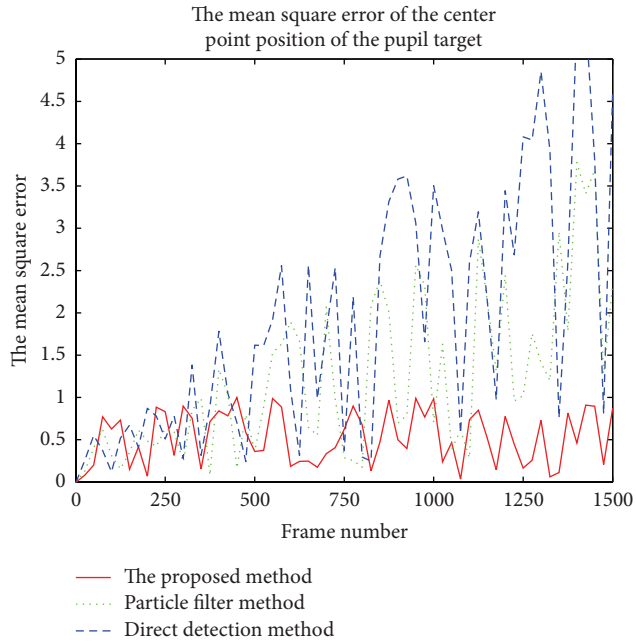


Figure 14: The RMSE of pupil center.

Table 2: The pupil parameters and RMSE based on three methods.

	Parameters of particle filtering (#216#651#1091#1106)	Parameters of detection (#513#690#732#1123)	Parameters of proposed method (#46#221#239#1140)
User 1 (Figure 14)	X: 564, Y: 434	X: 557, Y: 434, major: 9, minor: 10	X: 542, Y: 429, major: 9, minor: 10
	X: 524, Y: 434	X: 512, Y: 441, major: 8, minor: 10	X: 532, Y: 438, major: 8, minor: 10
	X: 576, Y: 454	X: 599, Y: 467, major: 9, minor: 9	X: 568, Y: 454, major: 9, minor: 9
	X: 468, Y: 376	X: 496, Y: 382, major: 9, minor: 10	X: 475, Y: 375, major: 9, minor: 10
User 2 (Figure 14)	X: 476, Y: 213	X: 452, Y: 219, major: 7, minor: 8	X: 492, Y: 206, major: 7, minor: 8
	X: 490, Y: 300	X: 497, Y: 284, major: 7, minor: 8	X: 471, Y: 298, major: 5, minor: 5
	X: 603, Y: 220	X: 510, Y: 290, major: 7, minor: 7	X: 557, Y: 259, major: 7, minor: 7
	X: 322, Y: 234	X: 332, Y: 225, major: 7, minor: 8	X: 386, Y: 203, major: 7, minor: 8
User 3 (Figure 14)	X: 674, Y: 340	X: 567, Y: 282, major: 9, minor: 10	X: 132, Y: 371, major: 8, minor: 8
	X: 692, Y: 332	X: 469, Y: 277, major: 9, minor: 9	X: 375, Y: 292, major: 7, minor: 9
	X: 636, Y: 192	X: 494, Y: 229, major: 8, minor: 9	X: 368, Y: 305, major: 5, minor: 8
	X: 569, Y: 321	X: 565, Y: 328, major: 8, minor: 9	X: 567, Y: 331, major: 8, minor: 9
User 1 variance (Figure 14)	d: 21.954, d: 14.318	d: 16.279, d: 21.514	d: 6.708, d: 5.657
	d: 21.024, d: 5.318	d: 24.012, d: 15.524	d: 4.472, d: 6.708
User 2 variance (Figure 14)	d: 19.106, d: 15.811	d: 43.382, d: 24.597	d: 6.325, d: 5.000
	d: 64.815, d: 49.930	d: 49.930, d: 51.264	d: 6.245, d: 4.123
User 3 variance (Figure 14)	d: 15.624, d: 21.128	d: 12.258, d: 17.174	d: 5.886, d: 5.241
	d: 18.854, d: 12.853	d: 24.263, d: 10.254	d: 4.236, d: 5.851
User 1 (variance/1500 frames)	d: 15.325	d: 20.832	d: 6.254
User 2 (variance/1500 frames)	d: 25.248	d: 42.214	d: 6.432
User 3 (variance/1500 frames)	d: 17.115	d: 15.980	d: 5.304

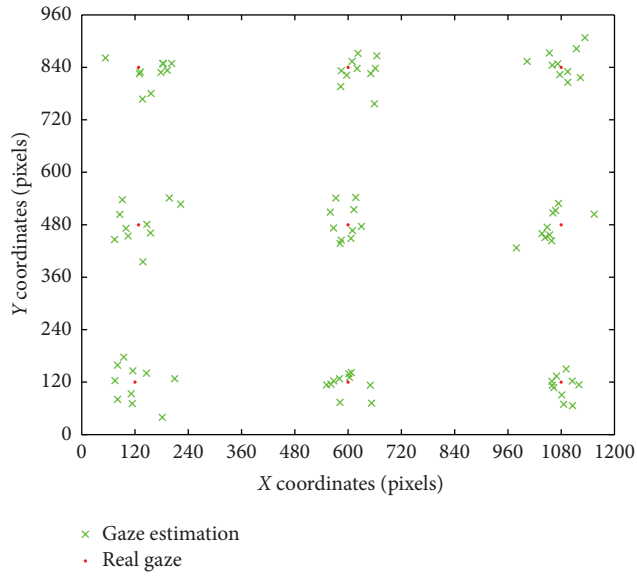


Figure 15: The gaze estimation results.

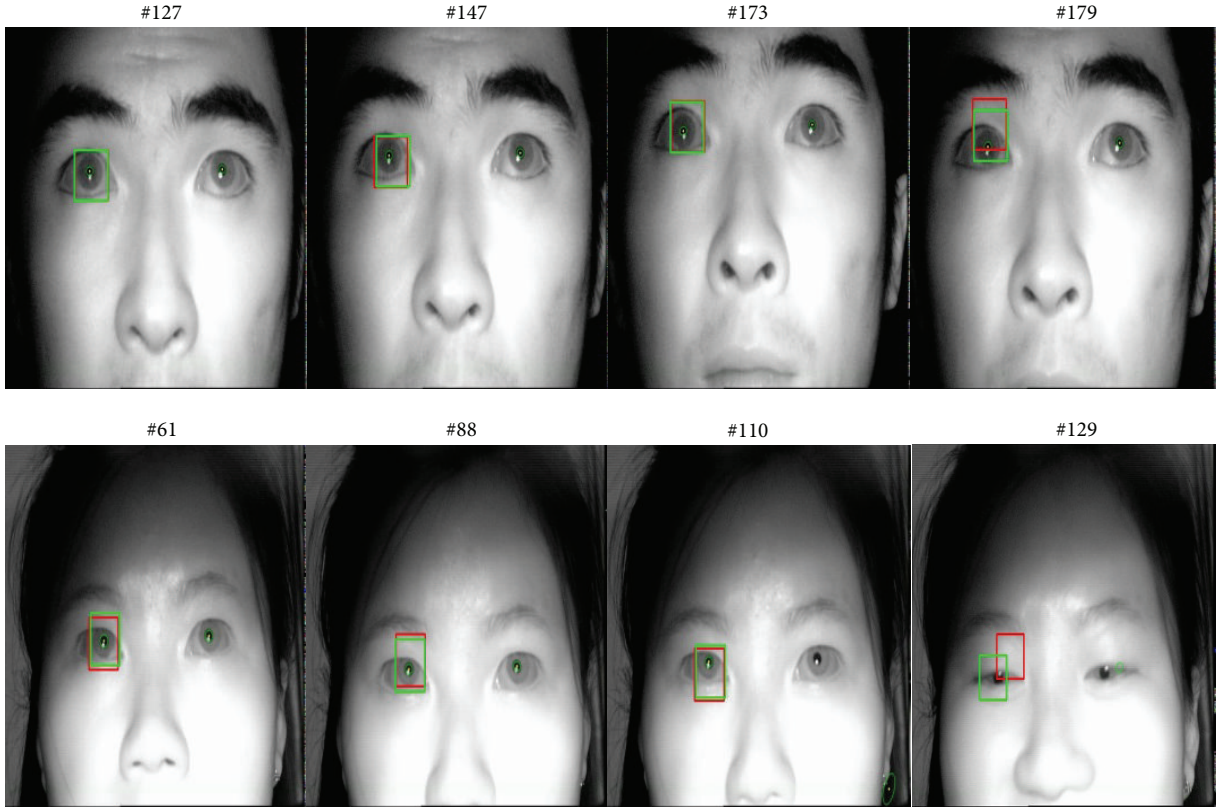


Figure 16: Experimental results of the proposed method.

results on the screen of user 1 based on the proposed method in gaze tracking system described in Figure 2.

This paper proposes a framework for target tracking by using particle filter combining with Kalman filtering. It not only meets requirement of the rapid target location with particle filter, but also estimates the parameters of the target with Kalman filtering. The entire framework of this algorithm achieves the rapid target location and accurate detection of pupil parameters. Currently, many literatures also proposed some portfolio tracking methods. In the literature [17], pupil is located with mean shift method firstly, and then the algorithm updates the tracking parameters with Kalman filtering. The literature [34] tracks the target with particle filter combined with mean shift. In order to verify the effectiveness of the method of particle filter combining with Kalman filtering proposed in this paper, we compare our method with the method of the literature [17], the literature [34], and the method of particle filtering. We capture 2 groups of image sequences which includes about 500 frames in each group. We choose 235 frames from each group including normal image and blinking image. The four methods mentioned above are used for tracking pupil target. The tracking results of frames 127, 147, 173, 179 and 61, 88, 110, 129 from each image sequence are shown in Figures 1, 2, 3, and 4. In Figures 1 and 2, the green rectangle is the final results of our method, which is updated by Kalman filtering. The red rectangle is the results before Kalman filtering updating. Figures 3 and 4 are the results by using particle filtering and using the method of particle filtering combined with mean shift, respectively. Figures 1 and 3 show that when the tracking error is relatively larger by using particle filter, our method can obtain the more accurate results. During the tracking process, the robustness of our method is also better. Figure 2 shows the results of Kalman filtering combined with mean shift, which use the method mean shift firstly for target location and then use Kalman filtering to update the parameters of the tracking target. For the pupil tracking, the results of the method of Kalman filtering combined with mean shift are not better than the method we proposed but are better than the other two methods. For more complex target tracking problem, the effect of the proposed tracking method where particle filtering is combined with Kalman filtering will be far superior to the method where Kalman filtering is combined with mean shift (see Figures 16, 17, 18, and 19). The results of the method of particle filtering combinedwith mean shift are shown in Figure 4. In this method the local peaks of all particles are gotten by using mean shift; then the tracking process is calculated together with the original particles and processed particles. But the experimental results show that the tracking accuracy of this method is not very nice. This is because some particles have offset the original particle states, which make the calculated location farther away from true position. But the number of particles used in this method is less than that of other methods, in which only 50 particles can achieve better results while the number of other methods is 200. The results of these tracking methods are shown in

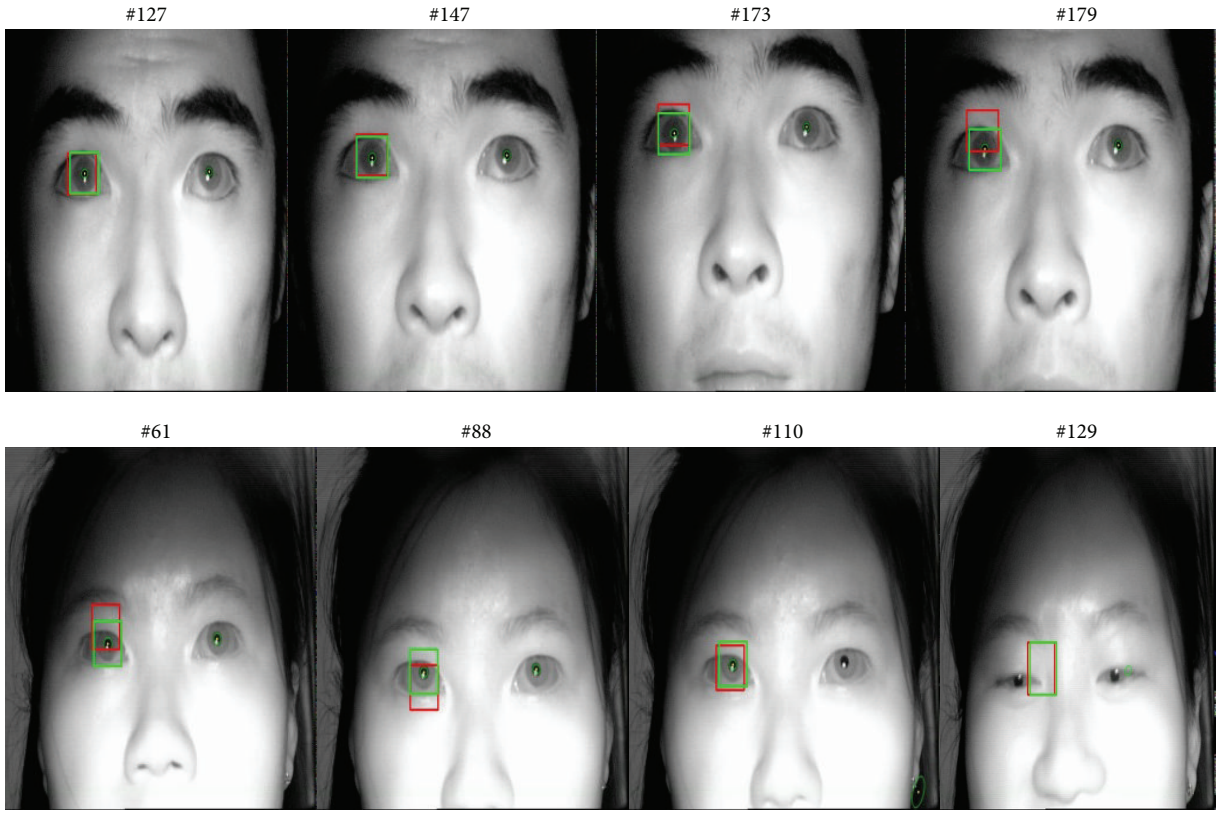


Figure 17: Experimental results of the method of Kalman filtering combined with mean shift.

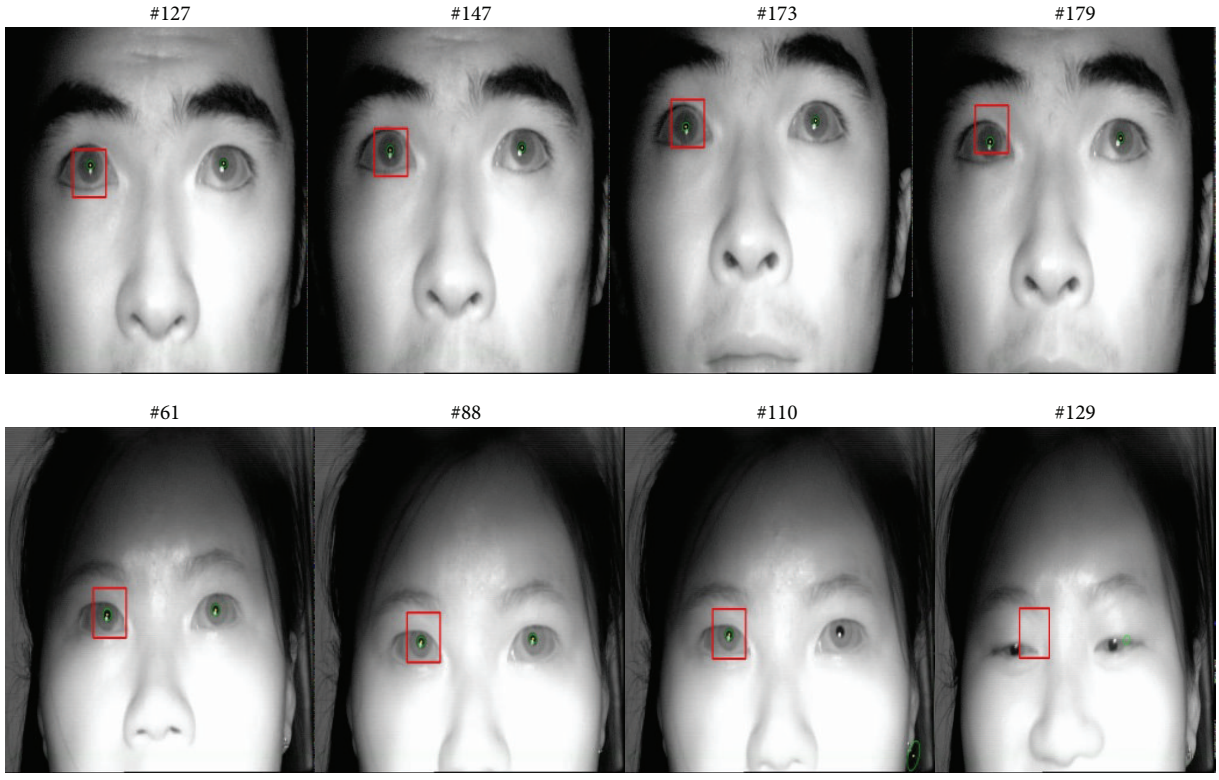


Figure 18: Experimental results of particle filtering.

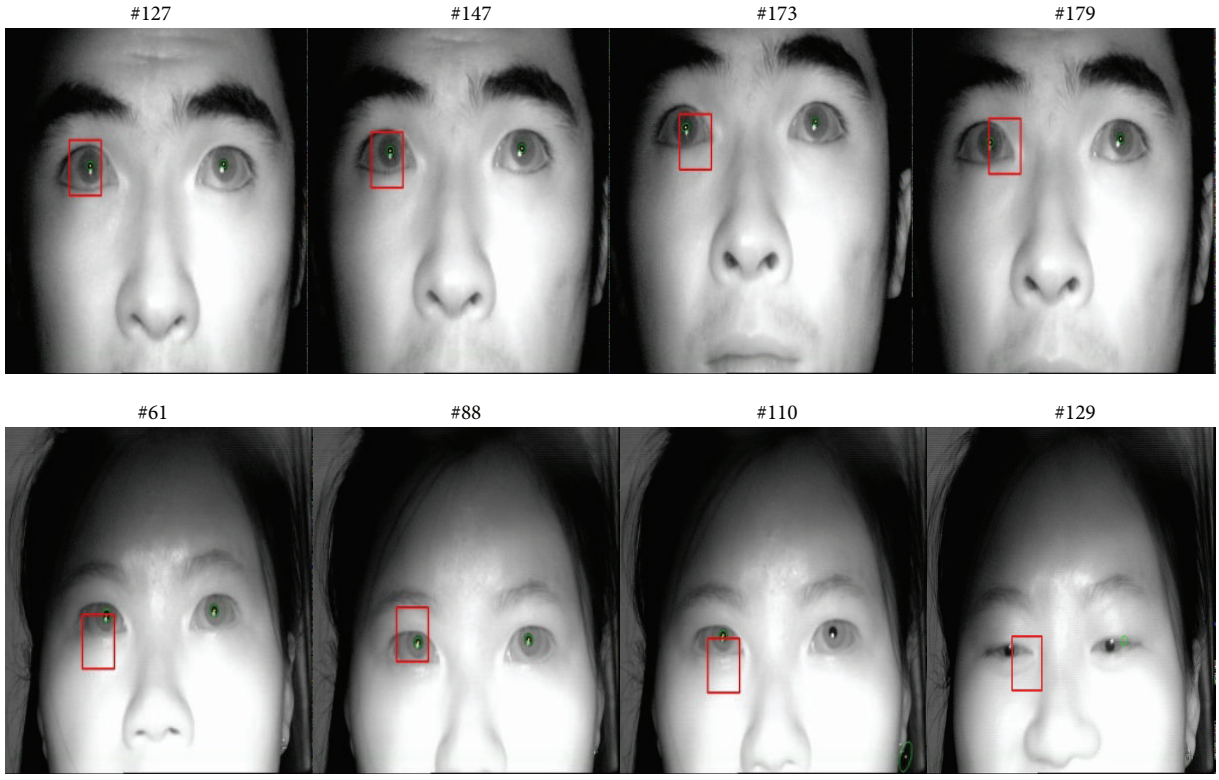


Figure 19: Experimental results of the method of particle filter combined with mean shift.

Table 3: Comparison of pupil tracking results and positioning time.

Parameter	Kalman + mean shift		Particle filter		Particle + mean shift		Our method	
Image classification	Normal	Wink	Normal	Wink	Normal	Wink	Normal	Wink
Image number	224	11	224	11	224	11	224	11
Accurate tracking number	218	2	199	6	169	7	219	8
Success rate (%)	97.32	18.18	88.84	54.55	75.44	63.63	97.77	72.73
The average success rate (%)	93.62		87.23		74.89		96.59	
Positioning time (ms/frame)	38		51		49		46	

Table 3. We can see that the results of the proposed method are better than those of other combination methods [17, 34].

5. Conclusions

In this paper, we proposed a method of pupil tracking in a gaze tracking system based on the combination of particle filtering and Kalman filtering when we oriented to fast and accurate pupil target detection using an active infrared source gaze tracking system. In order to make the color feature of the pupil more prominent, we synthesize triple-channel color image by using the bright pupil image, dark pupil image, and difference pupil image. Particle filtering is applied for fast pupil tracking by using triple-channel color image. On the basis of the tracking results of particle filtering, we use Kalman filtering to attain more accurate pupil parameters in difference image. Experimental results and analysis verify and illustrate the effectiveness of the proposed method.

Conflict of Interests

The authors declare that there is no conflict of interests regarding the publication of this paper.

Acknowledgments

The work is supported by Beijing Key Discipline Development Program (no. XK100080537), the Beijing Natural Science Foundation (4122050), and the Opening Project of Key Laboratory of Operation Safety Technology on Transport Vehicle, Ministry of Transport, China.

References

[1] T. Duchowski, *Eye Tracking Methodology: Theory and Practice*, Springer, New York, NY, USA, 2003.

[2] R. J. K. Jacob, "The use of eye movements in human computer i nteraction techniques: what you look at is what you get," *ACM Transactions on Information Systems*, vol. 9, pp. 152–169, 1991.

[3] K. R. Park, J. J. Lee, and J. Kim, "Gaze position detection by computing the three dimensional facial positions and motions," *Pattern Recognition*, vol. 35, no. 11, pp. 2559–2569, 2002.

[4] Z. Zhu and Q. Ji, "Novel eye gaze tracking techniques under natural head movement," *IEEE Transactions on Biomedical Engineering*, vol. 54, no. 12, pp. 2246–2260, 2007.

[5] Y. J. Ko, E. C. Lee, and K. R. Park, "A robust gaze detection method by compensating for facial movements based on corneal specularities," *Pattern Recognition Letters*, vol. 29, no. 10, pp. 1474–1485, 2008.

[6] C. Dong-Chan and K. Whoi-Yul, "Long-range gze tracking system for large movements," *IEEE Transactions on Biomedical Engineering*, vol. 60, no. 12, pp. 3432–3430, 2013.

[7] H. Lee, S. Y. Lim, I. Lee, J. Cha, D. Cho, and S. Cho, "Multi-modal user interaction method based on gaze tracking and gesture recognition," *Signal Processing: Image Communication*, vol. 28, no. 2, pp. 114–126, 2013.

[8] C. H. Morimoto, D. Koons, A. Amir, and M. Flickner, "Pupil detection and tracking using multiple light sources," *Image and Vision Computing*, vol. 18, no. 4, pp. 331–335, 2000.

[9] Y. Ebisawa, "Improved video-based eye-gaze detection method," *IEEE Transactions on Instrumentation and Measurement*, vol. 47, no. 4, pp. 948–955, 1998.

[10] A. de Santis and D. Iacoviello, "Robust real time eye tracking for computer interface for disabled people," *Computer Methods and Programs in Biomedicine*, vol. 96, no. 1, pp. 1–11, 2009.

[11] Q. Ji and X. J. Yang, "Real time visual cues extraction for monitoring driver vigilance," *Computer Vision Systems*, vol. 2095, pp. 107–112, 2001.

[12] Z. Lin, Y. Zhou, and W. An, "Improved multitarget track-before-detect using probability hypothesis density filter," *Journal of Infrared and Millimeter Waves*, vol. 31, no. 5, pp. 475–480, 2012.

[13] F. Lehmann, "Recursive bayesian filtering for multitarget track-before-detect in passive radars," *IEEE Transactions on Aerospace and Electronic Systems*, vol. 48, no. 3, pp. 2458–2480, 2012.

[14] S. J. Davey, N. J. Gordon, and M. Sabordo, "Multi-sensor track-before-detect for complementary sensors," *Digital Signal Processing*, vol. 21, no. 5, pp. 600–607, 2011.

[15] A. Zhu, C. Zhang, L. Lv, Y. Liu, and A. Duan, "Dual-module data fusion for detect before track based particle filter," in *Proceedings of the 2nd International Conference on Intelligent Control and Information Processing (ICICIP '11)*, pp. 729–731, IEEE, Harbin, China, July 2011.

[16] Z. Zhu, F. Kikuo, and Q. Ji, "Real-time eye detection and tracking under various light conditions," *Eye Tracking Research & Applications*, vol. 10, pp. 139–144, 2002.

[17] Z. Zhu and Q. Ji, "Robust real-time eye detection and tracking under variable lighting conditions and various face orientations," *Computer Vision and Image Understanding*, vol. 98, no. 1, pp. 124–154, 2005.

[18] D. W. Hansen and R. I. Hammoud, "An improved likelihood model for eye tracking," *Computer Vision and Image Understanding*, vol. 106, no. 2-3, pp. 220–230, 2007.

[19] D. W. Hansen and A. E. C. Pece, "Eye tracking in the wild," *Computer Vision and Image Understanding*, vol. 98, no. 1, pp. 155–181, 2005.

[20] T. Liu and S. Zhu, "Eyes detection and tracking based on entropy in particle filter," in *Proceeding of the 5th International Conference on Control and Automation (ICCA '05)*, vol. 2, pp. 1002–1007, IEEE, Budapest, Hungary, June 2005.

[21] R. Campos, C. Santos, and J. Sequeira, "Eye tracking system using particle filters," in *Proceedings of the IEEE 3rd Portuguese Meeting in Bioengineering (ENBENG '13)*, pp. 1–4, IEEE, Braga, Portugal, February 2013.

[22] K. Nummiaro, E. Koller-Meier, and L. van Gool, "An adaptive color-based particle filter," *Image and Vision Computing*, vol. 21, no. 1, pp. 99–110, 2003.

[23] Z. Zhu, *Paritcle Filtering and Its Application*, Science Press, Beijing, China, 2010.

[24] M. Morshidi and T. Tjahjadi, "Gravity optimised particle filter for hand tracking," *Pattern Recognition*, vol. 47, pp. 194–207, 2014.

[25] Z. Chen, *Bayesian Filtering: From Kalman Filers to Particle Filters, and Beyond*, McMaster University, Hamilton, Canada, 2003.

[26] S. Haykin, Ed., *Kalman Filtering and Neural Networks*, John Wiley & Sons, New York, NY, USA, 2001.

[27] C. Zhang, J. Chi, Z. Zhang, and Z. Wang, "Research on eye tracking method in gaze tracking system," *Acta Automatica Sinica*, vol. 36, no. 8, pp. 1051–1061, 2010.

[28] V. S. Zaritskii, V. B. Svetnik, and L. I. Shimelevich, "Monte-Carlo technique in problems of optimal information processing," *Automation and Remote Control*, vol. 36, no. 3, pp. 2015–2022, 1975.

[29] N. J. Gordon, D. J. Salmond, and A. F. M. Smith, "Novel approach to nonlinear/non-gaussian Bayesian state estimation," *IEE Proceedings F: Radar and Signal Processing*, vol. 140, no. 2, pp. 107–113, 1993.

[30] G. R. Bradsik, "Real time face and object tracking as a component of a perceptual user interface," in *Proceedings of the 4th Workshop on Application of Computer Vision*, pp. 214–219, IEEE Computer Society Press, Washington, DC, USA, 1998.

[31] R. Valenti and T. Gevers, "Accurate eye center location and tracking using isophote curvature," in *Proceedings of the 26th IEEE Conference on Computer Vision and Pattern Recognition (CVPR '08)*, pp. 1–8, Anchorage, Alaska, USA, June 2008.

[32] Y. Chu, H. Mi, and Z. Ji, "Image registration based on weighted histogram for digital subtraction angiography," *Acta Electronica Sinica*, vol. 36, no. 7, pp. 1344–1348, 2008.

[33] H. You, W. Guohong, L. Dajin, and P. Yingning, *Multi-sensor data fusion and its application [M.S. thesis]*, Press of Electronics Industry, 2000.

[34] C. Shan, T. Tan, and Y. Wei, "Real-time hand tracking using a mean shift embedded particle filter," *Pattern Recognition*, vol. 40, no. 7, pp. 1958–1970, 2007.

Hindawi Publishing Corporation
Mathematical Problems in Engineering
Volume 2014, Article ID 568034, 20 pages
http://dx.doi.org/10.1155/2014/568034

Research Article

Self-Similarity Based Corresponding-Point Extraction from Weakly Textured Stereo Pairs

Min Mao,[1] Kuang-Rong Hao,[1,2] and Yong-Sheng Ding[1,2]

[1] *College of Information Sciences and Technology, Donghua University, Shanghai 201620, China*
[2] *Engineering Research Center of Digitized Textile & Fashion Technology, Donghua University, Ministry of Education, Shanghai 201620, China*

Correspondence should be addressed to Kuang-Rong Hao; krhao@dhu.edu.cn and Yong-Sheng Ding; ysding@dhu.edu.cn

Received 13 March 2014; Revised 15 August 2014; Accepted 15 August 2014; Published 3 September 2014

Academic Editor: Yi Chen

For the areas of low textured in image pairs, there is nearly no point that can be detected by traditional methods. The information in these areas will not be extracted by classical interest-point detectors. In this paper, a novel weakly textured point detection method is presented. The points with weakly textured characteristic are detected by the symmetry concept. The proposed approach considers the gray variability of the weakly textured local regions. The detection mechanism can be separated into three steps: region-similarity computation, candidate point searching, and refinement of weakly textured point set. The mechanism of radius scale selection and texture strength conception are used in the second step and the third step, respectively. The matching algorithm based on sparse representation (SRM) is used for matching the detected points in different images. The results obtained on image sets with different objects show high robustness of the method to background and intraclass variations as well as to different photometric and geometric transformations; the points detected by this method are also the complement of points detected by classical detectors from the literature. And we also verify the efficacy of SRM by comparing with classical algorithms under the occlusion and corruption situations for matching the weakly textured points. Experiments demonstrate the effectiveness of the proposed weakly textured point detection algorithm.

1. Introduction

Local interest points can be used for many applications, such as video analyzing, object detection, localization, and identification. Techniques which only use images to reconstruct 3D scenes are now the most popular of its applications. Since many algorithms based on the diversity of concepts such as graph cuts [1] and minimal path search [2] can only handle short-baseline stereo matching, that is, they cannot be used at the wide-baseline situation, on the other hand, gradient detectors can be used for wide-baseline matching well; in general, the process for matching the discrete image points can be divided into three main steps. First, extracting the "interest points" from each image, such as T-junctions, corners, and the point detector should have the property of repeatability, which guarantee finding the same physical interest points under different viewing conditions. Next, each interest point can be represented by a feature vector through the descriptor, which should also be distinctive, finally, matching the descriptor vectors between different images.

Almost all of detectors are based on the gradient map of image; for example, the Harris corner detector [3] is based on the second moment matrix, which describes the gradient distribution in a local neighborhood of a point in image, but corners detected by this method are not scale invariant. Mikolajczyk and Schmid [4] proposed two scale-invariant methods, that is, Harris-Laplace and Hessian-Laplace, which are based on the concept of automatic scale selection [5], and the location is selected by Harris measure or the determinant of the Hessian matrix; scale is selected by Laplacian; Lowe [6] speeds up the above methods by using the difference of Gaussians (DoG) to approximate the Laplacian of Gaussians (LoG). There are lots of different detectors which have been proposed in the literature [7–10]. However, problems of methods based on interest point still remain, such as image

blurring, magnification, and illumination; one of the most serious weaknesses between these is that these methods could not get point on the low texture areas. Since these areas can be defined by its gradient below some constant thresholds, it is difficult to extract points on them by classical interest-point detectors.

As a guiding principle for describing shapes this characteristic has a rich history. Animals, man-made objects, and plants are usually with symmetry characteristic. Many techniques have been proposed to analyze this characteristic. For example [11] describes an algorithm to segment objects into terms of points, line segments, and circles. Loy and Zelinsky [12] use the local radial symmetry to highlight points of interest within a scene, which need not consider the contribution of a local neighborhood to a central pixel. In this paper we focus on the characteristic of strong self-similarity. The approach proposed in [13] inspires us that the region of weakly textured area has the characteristic of strong self-similarity; hence the weakly textured points can be located by this characteristic; moreover the locations with self-similar structure of local pattern are also distinguishable in different images. Our goal is to develop both a detector and matching algorithm for getting corresponding-point in weakly textured regions under wide-baseline situation. Our work in this paper can be divided into three parts, namely, candidates of weakly textured point detection, refining the set of candidates, and weakly textured point matching algorithm based on sparse representation.

In the detecting part, the proposed detector is based on the entities: circumferences and radii; it sums the value on circumferences and radii; the information from image is very first level of data processing, which is contrary to gradient-based methods; hence, for giving the scale invariant as the gradient-based methods we introduce a mechanism for automatic radial selection. The following two approaches can be identified; the first one is used to obtain the symmetry maps for each pixel in different radius scales, which is also used in [14]. The second approach selects the radius scale for each pixel.

In the refinement part, the set of weakly textured points can be refined by computing the gradient magnitude in scale-space [15], which is widely used in image processing community, such as feature description, point detection, and image structure analysis. Here we use this theory to measure the texture strength and for reducing the points which do not belong to weakly textured point; we also propose a threshold selection mechanism.

In the point matching part, the correspondence relationship of points which contain the same scene in two images will be determined. The matching between two points set is a one-to-one mapping between the points in these two sets. There are many algorithms for computing the similarity between two point features, such as zero-mean normalized cross correlation (ZNCC) [16] and Hausdorff distance [17], which many algorithms have been suggested for. We here introduced the sparse representation concept [18] for weakly textured point matching, which has been used for human face recognition in [19]. This approach considers all possible supports (here is the set of weakly textured points in the second image) and can automatically choose the minimal number of samples needed to represent each feature of point in the first image.

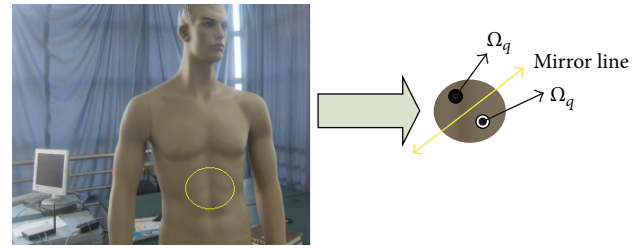


Figure 1: Example of a weakly textured region in human model.

The approach proposed in this work gives a new detector for extracting the weakly textured points from image in which the objects have weakly textured characteristic. Moreover, since the detector is based on radial symmetry, it is not sensitive to variations in image illumination. In the experiment for detectors, the results obtained on image sets of different image sets under different types of geometric and photometric transformations show that the extracted points by this detector are the complement of classical detectors and high robustness with intraclass variations. In the point matching algorithms test experiment, the method based on sparse representation outperforms other algorithms under the pixel corruption and block occlusion situations, respectively, and the results also show that the extracted weakly textured points are also distinguishable from locations; finally we use our algorithm to promote the quasi-dense matching algorithm which was proposed in [20] to verify the proposed approach under wide-baseline situation.

This paper is organized as follows: in Section 2 the method is proposed to detect the candidate of weakly textured regions. In Section 3, we sketch out a mechanism for points refinement via scale-space selection theory. In Section 4, the sparse representation theory is used for matching weakly textured point. In Section 5, the proposed method is tested in accordance with two protocols [21, 22] for evaluation of local region detectors and matching algorithm. Section 6 provides concluding remarks and possible extensions of the proposed method.

2. Weakly Textured Feature Points Detection

Given an image I with the weakly textured region Ω_{WT} and texture region Ω_T, it is clear that there has no corner-point or have few edge-points in the neighbor region of the points in Ω_{WT} (see Figure 1). Let $I(x, y)$ denote the gray value of image; then the points' characteristic in the region Ω_{WT} can be represented by its first- and second-order partial derivatives as

$$I_x, I_y, I_{xx}, I_{yy}, I_{xy}, I_{yx} \approx 0, \quad \{(x, y) \in \Omega_{\mathrm{WT}}\}. \tag{1}$$

For the above mentioned, the detectors based on the gradient feature such as the Hessian, Harris, and SIFT detectors could not be used for detecting these points. So

these points should be detected in different ways. As shown in Figure 1, the region formed by weakly textured pixel and its neighborhoods can be segmented in many fragments and the pixel's gray value distribution in each fragment is similar.

Let $I(x)$ be the intensity value at location; Ω_p denotes the reference fragment in the image; then the self-similar fragment Ω_p can be measured by the normalized correlation coefficient as follows:

$$\mathrm{ncc}(P,T) = \frac{\sum_i \left(I(x_i) - \bar{I}\right)\left(I(T(x_i)) - \bar{I}\right)}{\sqrt{\left(\sum_i \left(I(x_i) - \bar{I}\right)^2\right)\left(\sum_i \left(I(T(x_i)) - \bar{I}\right)^2\right)}}, \quad \{p \neq q \mid q = 1, 2, \ldots, N\}, \tag{2}$$

where $I(T(x)) = a + bI(x)\ \forall x \in \Omega_p$.

$\bar{I}$ denotes the average intensity value of fragment Ω_p. The purpose of the formula $I(x_i) - \bar{I}$ is to reduce the influence caused by intensity; $T(\cdot)$ denotes a geometrical transformation.

From Figure 1, it is clear that the weakly textured region has strong mirror symmetry; let point $p = (r, \theta)$ be represented in polar coordinates, let $\varphi \in [0, \pi)$ denote the mirror line orientation, and then the symmetric point of p about the mirror line can be represented as $p' = (r, 2\varphi - \theta)$. For measuring the mirror symmetry about the region Ω_p, it should fulfill all mirror line orientation $\varphi \in [0, \pi)$; here the symmetry of region Ω_p can be obtained as follows [23]:

$$S_{w\text{-texture}}\left(\Omega_p\right) = \frac{1}{N}\sum_{i=0}^{N-1} \mathrm{ncc}\left(\Omega_p, T\left(\Omega_p, \varphi_i\right)\right), \tag{3}$$

where $T(\cdot)$ denotes the symmetry transformation function, and it transforms the Ω_p to its symmetric region about the mirror line orientation $\varphi_i \in \{\varphi_0, \varphi_1, \ldots, \varphi_{N-1}\}$, $\Delta\varphi = \pi/N$, and $\varphi_i = i \times \Delta\varphi$. It is easy to prove that the $\max(S_{w\text{-texture}}(\Omega_p)) = 1$ and $\min(S_{w\text{-texture}}(\Omega_p)) = 0$, $\Omega_p \in I$.

2.1. Same Radial for $S_{w\text{-texture}}$. In the example shown in Figure 2 the strength of $S_{w\text{-texture}}$ with the same radial in different images is affected by the similarity of its region, as we can see that the strong texture region has small value of $S_{w\text{-texture}}$; in contrast, the value of $S_{w\text{-texture}}$ in weakly textured region is close to 1. So $S_{w\text{-texture}}$ can be used for detecting the weakly textured region. However the middle-down part of Figure 2 shows that the region with strong similarity but not weakly textured region has high value of $S_{w\text{-texture}}$; this case indicates that the weak texture cannot be detected by $S_{w\text{-texture}}$ only. And in Section 3, we will solve this problem by the differential expressions at the center point of region Ω_p.

2.2. Radial Selection for $S_{w\text{-texture}}$. According to the above discussion, we use $S_{w\text{-texture}}$ as our weakly textured detector (WTD); however, there are many factors which can affect the performance of WTD, including the structure complexity distributed within the local region, variety illumination for the local region, and radials chosen by WTD. Here we investigate the performance at which WTD measures region's texture strength with different radials.

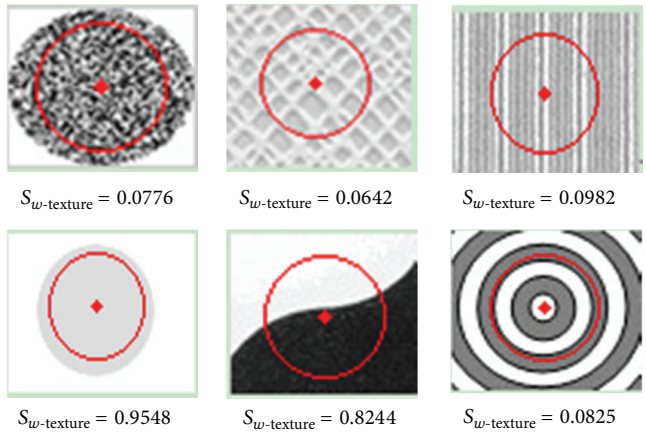


FIGURE 2: $S_{w\text{-texture}}$ in different image structures with the same radius.

As shown in Figure 3, the texture strength is reflected by the intensity of the pixel in the similarity map; that is, the pixel with lower texture strength has higher intensity. When the radial change from small to large the intensity of the pixel with low texture strength will become lower, whereas the intensity of the pixel with high texture strength will become higher.

The points detect by WTD with different radials are shown in Figure 4. Obviously, the result is largely affected by the radials of the operator.

Here we proposed a mechanism for automatic radial selection. The radial selection mechanism should follow heuristic principles: first, the radial must be large enough to contain the weakly textured region; that is, the radial should extend the weakly textured region until the region bound close to the texture feature. Second, the points chosen by this mechanism should have the maxima value of $S_{w\text{-texture}}$ with its radial. Third, the mechanism should have robustness to small variations caused by fragments; that is, the radial can ignore small fragments in the weakly textured region.

As shown in Figure 5, $S_{w\text{-texture}}$ varies with the radius, and let S_R be the derivative of $S_{w\text{-texture}}$ with radius. In differential geometric terms, the fine-radius for each weakly textured points should simply satisfy $S_{RR} = 0$, which is the second-order derivative of $S_{w\text{-texture}}$. On the other hand, when considering the influence caused by neighboring fragments, we here use the following *γ-parameterized normalized derivative:* $\partial_{RR}(R^{\gamma} S_{w\text{-texture}}) = 0$. According to this formula the influence caused by the fragments can be eliminated under the fitted parameterized γ, with $\gamma = 0.5$. Figure 6 shows the results with this radius selection method.

3. Weakly Textured Points Filter for Fine Positioning

In Section 2 we have analyzed the weakly textured feature and use WTD with proposed adaptive radial mechanism to position the weakly textured point. However the problem

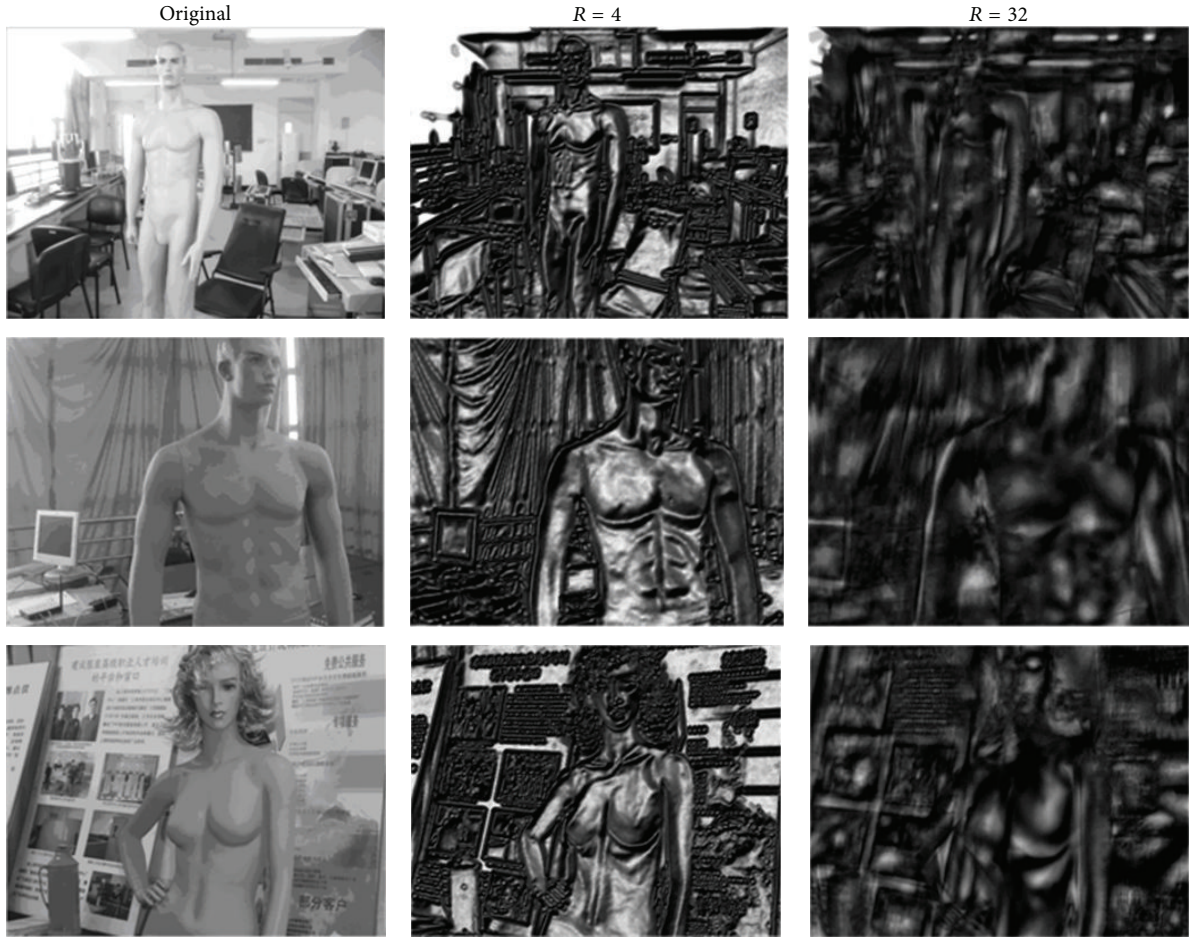


Figure 3: Similarity maps under radii $R = 4$ and $R = 32$.

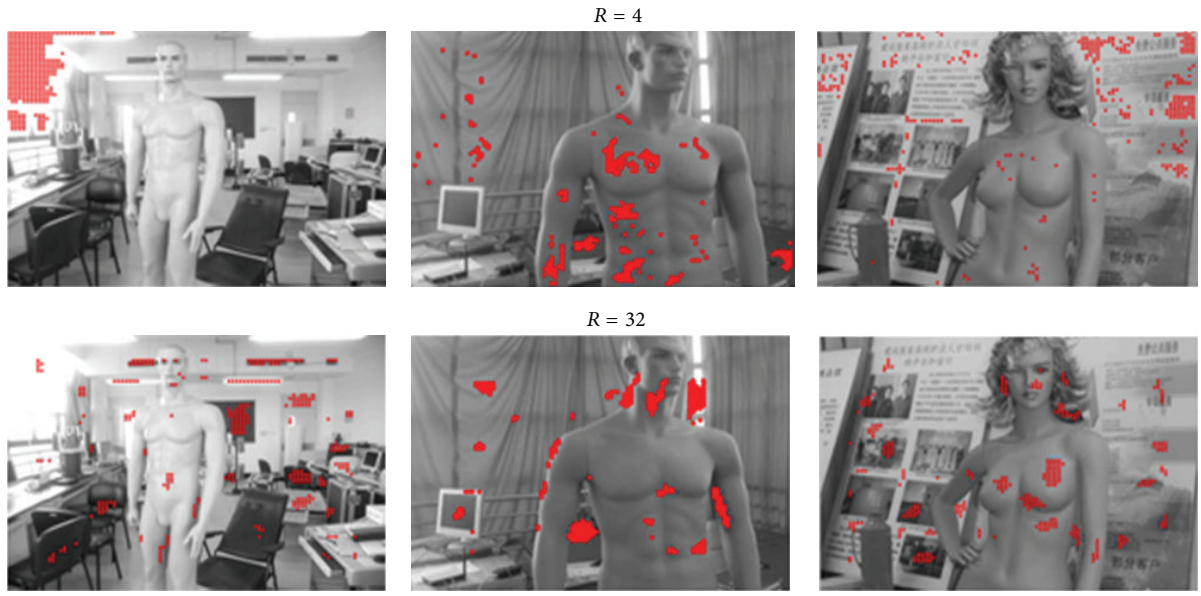


Figure 4: Extracted points under radii $R = 4$ and $R = 32$.

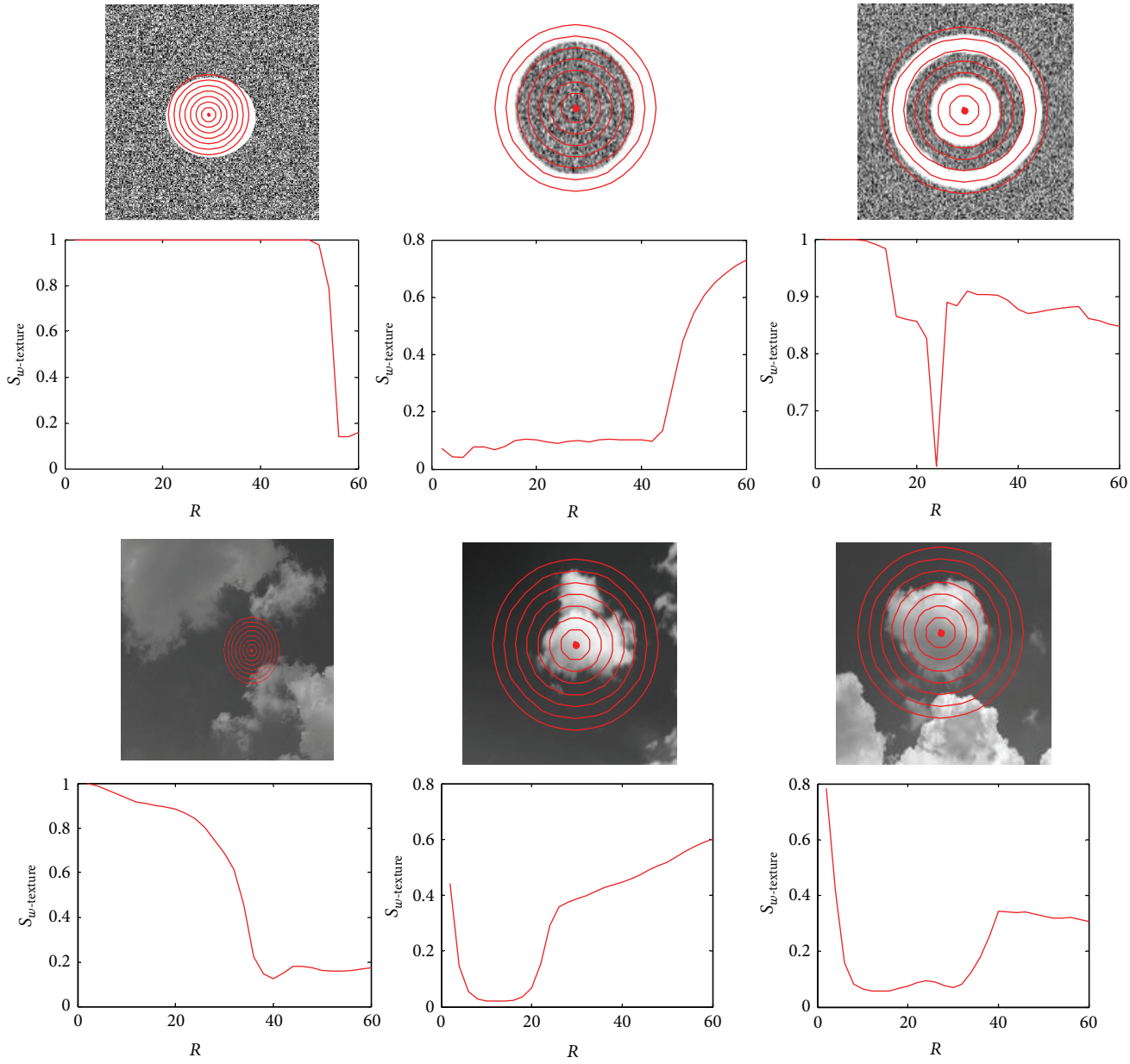


FIGURE 5: $S_{w\text{-texture}}$ varies on a scale from $R = 2$ to $R = 60$.

FIGURE 6: Radius selected for weakly textured points.

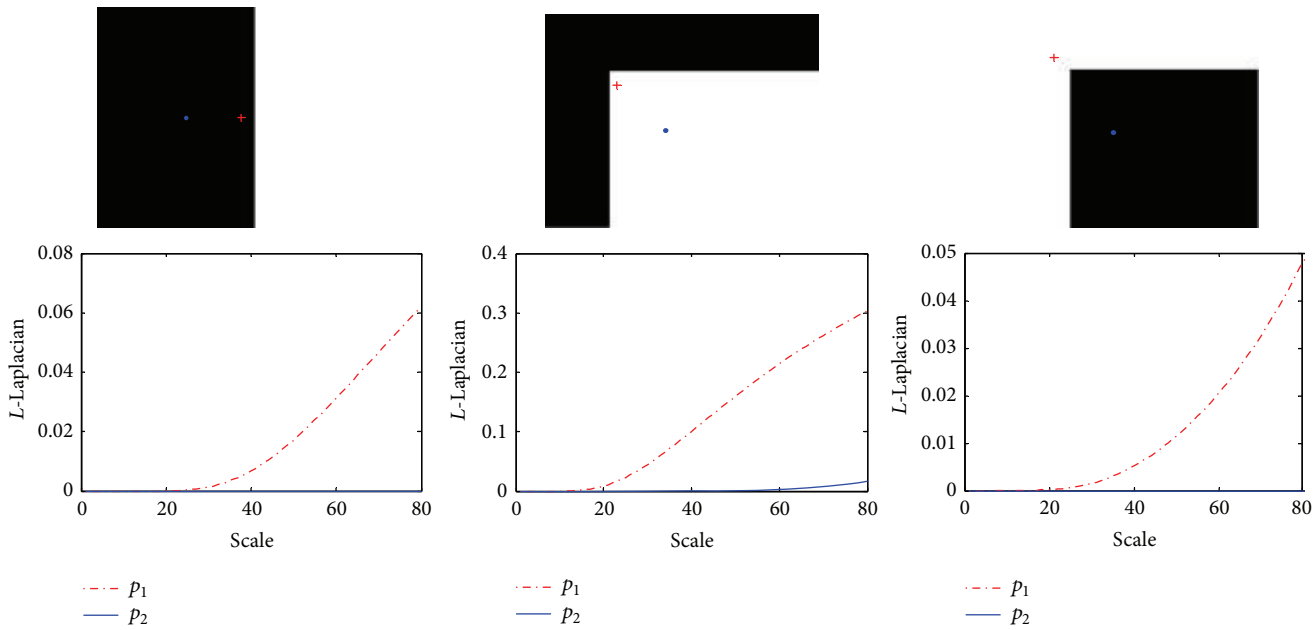


FIGURE 7: L_{grad} varies on scale-space from $t = 10$ to $t = 80$.

in the middle-down part of Figure 2 shows that the point has large value of $S_{w\text{-texture}}$ which is not the weakly textured point. On the other hand, the state-of-the-art edge detector corresponds to detecting points with the maximum gradient magnitude in the gradient direction, so edges can be detected in scale-space; here we extend these ideas to remove these points which do not have the weakly textured property.

3.1. Improve WTD in Scale-Space. When giving a 2D signal f, the definition of the scale-space representation $L : R^2 \times R_+ \rightarrow R$ is the solution of the diffusion equation [24]:

$$\partial_t L = \frac{1}{2}\nabla^2 L = \frac{1}{2}\left(\partial_{xx} + \partial_{yy}\right) L. \tag{4}$$

It can be proved $L(\cdot, t) = g(\cdot, t) * f(\cdot)$, where $g : R^2 \times R_+ \rightarrow R$ is the Gaussian kernel; that is,

$$g\left(x, y; t\right) = \frac{1}{2\pi t} e^{-(x^2+y^2)/(2t)}. \tag{5}$$

For removing the point with strong texture feature, we should detect its texture strength in its neighbor region; here we use the gradient magnitude to represent it. Based on the γ-*parameterized* normalized derivative concept in the scale-space, we will consider the following differential expression:

$$L_{\text{grad}}\left(\cdot; \sqrt{t}\right) = t^{\gamma} \left| \left(\frac{\partial L\left(\cdot; t\right)}{\partial x} \right)^2 + \left(\frac{\partial L\left(\cdot; t\right)}{\partial y} \right)^2 \right|. \tag{6}$$

As shown in Figure 7 when using this operator to measure the points' texture strength, it is clear that L_{grad} will change along with the scale t. Moreover the more close the point is to the edge the bigger the change to L_{grad} magnitude it has (the cross-point "+" in each picture denotes the one more close to the edge).

3.2. Point Texture Strength Based on Local Maximum over Scales of Derivative. As above mentioned, the point texture strength can be reflected by L_{grad} magnitude over scales; hence we can define it by the local maximum over scales of derivative, and here we use following step function which is the close-form theoretical analysis for edge models, and it also can be convenient for further analysis to instead the actual edges. Consider

$$f\left(x\right) = \begin{cases} 0 & \text{for } x < x_0 \\ 1 & \text{for } x \geq x_0. \end{cases} \tag{7}$$

On the other hand, Gaussian kernel is a local kernel which responds to near neighbor of the input variables; hence, we here restrict the kernel size to the interval $[-3\sqrt{t}, 3\sqrt{t}]$. Consider

$$g\left(x; t\right) = \begin{cases} \dfrac{1}{2\pi t} e^{-x^2/(2t)} & x \in \left[-3\sqrt{t}, 3\sqrt{t}\right] \\ 0 & \text{otherwise.} \end{cases} \tag{8}$$

By substitution from (7) and (8) in (6), we obtain the L_{grad} magnitude of $f(x)$ with the following equation:

$$L\left(x, t\right) = \begin{cases} 0 & x - x_0 \leq -3\sqrt{t} \\ \displaystyle\int_{-3\sqrt{t}}^{x-x_0} g\left(x'; t\right) dx' & -3\sqrt{t} < x - x_0 < 3\sqrt{t} \\ \displaystyle\int_{-3\sqrt{t}}^{3\sqrt{t}} g\left(x'; t\right) dx' & x - x_0 \geq 3\sqrt{t}. \end{cases} \tag{9}$$

According to above equation, magnitude L_{grad} of $f(x)$ can be represented as the following expression:

$$L_x(x,t) = \begin{cases} g(x - x_0, t) & |x - x_0| < 3\sqrt{t} \\ 0 & \text{otherwise,} \end{cases} \tag{10}$$

where $|x - x_0|$ represents the distance between x and the discontinuous point x_0 which is the point on the edge, and we use the symbol Δl instead of it. This formula shows that when $\Delta l < 3\sqrt{t}$, $L_{\text{grad}} \neq 0$, and as mentioned above, the local maximum over scales of derivative will be used as the point's texture strength. But from Figure 7 it can be shown that there is no guarantee for getting the local maximum of the derivative over scales. And the derivative of $L_x(x,t)$ can be gotten from formula (10) as follows:

$$\frac{\partial L_x}{\partial t} = \begin{cases} \dfrac{1}{2\sqrt{2\pi}} t^{-5/2} e^{-\Delta l^2/2t} \left(\Delta l^2 - t\right) & \Delta l < 3\sqrt{t} \\ 0 & \text{otherwise.} \end{cases} \tag{11}$$

It is clear that the local maximum of L_x depends on scale t; that is, if $t_0 = \Delta l^2$, then $\partial L_x/\partial t|_{t=t_0} = 0$. Hence the range of scale $[1, t_{\max}]$ is difficult to select; for example, when $\Delta l = 20$, then $t_{\max}$ must satisfy $t_{\max} \geq 400$, and when $t_{\max}$ is set too large, it will increase in computation complexity for finding the local maximum of L_x. Obviously, we want to find the way to ignore the influence caused by Δl; that is, t_0 is not proportional to Δl^2.

For this reason we need to transform $L_x(x,t)$ to the function in which local maximum does not depend on Δl^2; let $L'_x = p(t) \cdot L_x$; then $\partial L'_x/\partial t = (\partial p(t)/\partial t) L_x + p(t)(\partial L_x/\partial t)$; that is, the derivative of $\partial L'_x/\partial t$ is

$$\frac{\partial L'_x}{\partial t} = \begin{cases} \dfrac{1}{2\sqrt{2\pi}} t^{-5/2} e^{-\Delta l^2/2t} \left(2p(t)t^2 + p(t)\left(\Delta l^2 - t\right)\right) \\ \qquad \Delta l < 3\sqrt{t} \\ 0 \quad \text{otherwise.} \end{cases} \tag{12}$$

For the purpose of reducing the incremental of $t_{\max}$ when Δl increases, hence $p(t)$ here should be the decreasing function of scale t. On the other hand, despite the fact that $t_{\max}$ should not be proportional to Δl^2, it has to guarantee that the local maximum position t_0 increases with increasing Δl; that is, it should satisfy

$$\left.\begin{aligned} \frac{\partial L'_x(\Delta l_1, t)}{\partial t} &= 0, \quad (t = t_1) \\ \frac{\partial L'_x(\Delta l_2, t)}{\partial t} &= 0, \quad (t = t_2) \end{aligned}\right\} \Longrightarrow t_1 > t_2. \tag{13}$$

For the constraint to $p(t)$, it can be assumed that $p(t) = \alpha - t^\gamma/\beta$, where $\alpha, \beta > 0$, and it also should satisfy $\log \alpha\beta > \gamma \log t$ (this constraint can guarantee that if $t \geq 1$, then $L'_x > 0$); hence, for the above mentioned when $t \geq 1$, then $p(t)$, $p'(t) \neq 0$; if we insert $p(t)$ into (13) and give $\gamma = 1$, then the local maximum position t_0 is given by

$$\begin{aligned} \frac{\partial L'_x}{\partial t} = 0 \Longrightarrow t_0 &= \frac{-\left(\Delta l^2 + \alpha\beta\right) + \sqrt{\left(\Delta l^2 + \alpha\beta\right)^2 + 4\alpha\beta\Delta l^2}}{2} \\ &= \frac{2\alpha\beta\Delta l^2}{\left(\Delta l^2 + \alpha\beta\right) + \sqrt{\left(\Delta l^2 + \alpha\beta\right)^2 + 4\alpha\beta\Delta l^2}} \\ &= \frac{2\alpha\beta}{\left(1 + \alpha\beta/\Delta l^2\right) + \sqrt{\left(1 + \alpha\beta/\Delta l^2\right)^2 + 4\left(\alpha\beta/\Delta l^2\right)}}. \end{aligned} \tag{14}$$

It is clear that when $\alpha\beta/\Delta l^2 \to 0$, then $t_0 \to \alpha\beta$; moreover it is an increasing function of Δl; that is, it has the same monotonicity property with one in original function $\partial L_x/\partial t|_{t=t_0} = 0$. Figure 8 shows the result improved by $L'_x(x,t)$.

3.3. Threshold Selection for Weakly Textured Points. A method for threshold selection to remove points which are not weakly textured points is needed, if their local maximum of texture strength is beyond this threshold.

On the other hand, there is no a priori information that can be used for removing the point which has no weakly textured property, and to illustrate the method for threshold selection, here, let us consider the distribution of weakly textured points detected by the method in Section 2.

Figure 9 shows the weak points detected by WTD and the 3D histogram of their distribution in the image. Here, the image is divided into $M \times N$ fragments; that is, if I is the image, then $I = \{R_1, R_2, \ldots, R_{M \times N}\}$; according to this distribution we find that regions have more concentrated distribution than others which are weakly textured regions. It also indicates that the local maximum texture strength of points distributed in these regions can be used to detect the threshold.

Through the above result, the properties of weakly textured region can be assumed as follows.

(i) The points which are detected by WTD in the region with weakly textured feature have more concentrated distribution than other regions.

(ii) The mean of local maximum texture strength in the weakly textured region is lower than that in other regions.

An important consequence of these properties is that it allows the threshold to vary with different images. We here use the Otsu threshold algorithm to segment the point sets detected by WTD into two classes, that is, the sets belonging to weakly textured region and the sets belonging to other regions. Let T_O be the threshold defined by Otsu threshold algorithm; $N(R_i)$ is the number of the points in fragment R_i (as shown above, the image is divided into $M \times N$ fragments), and according to property (i), if $N(R_i) \geq T_O$, then fragment R_i is selected as the part of weakly textured region which will be used for determining the threshold of texture strength later. On the other hand, when the weakly textured region is

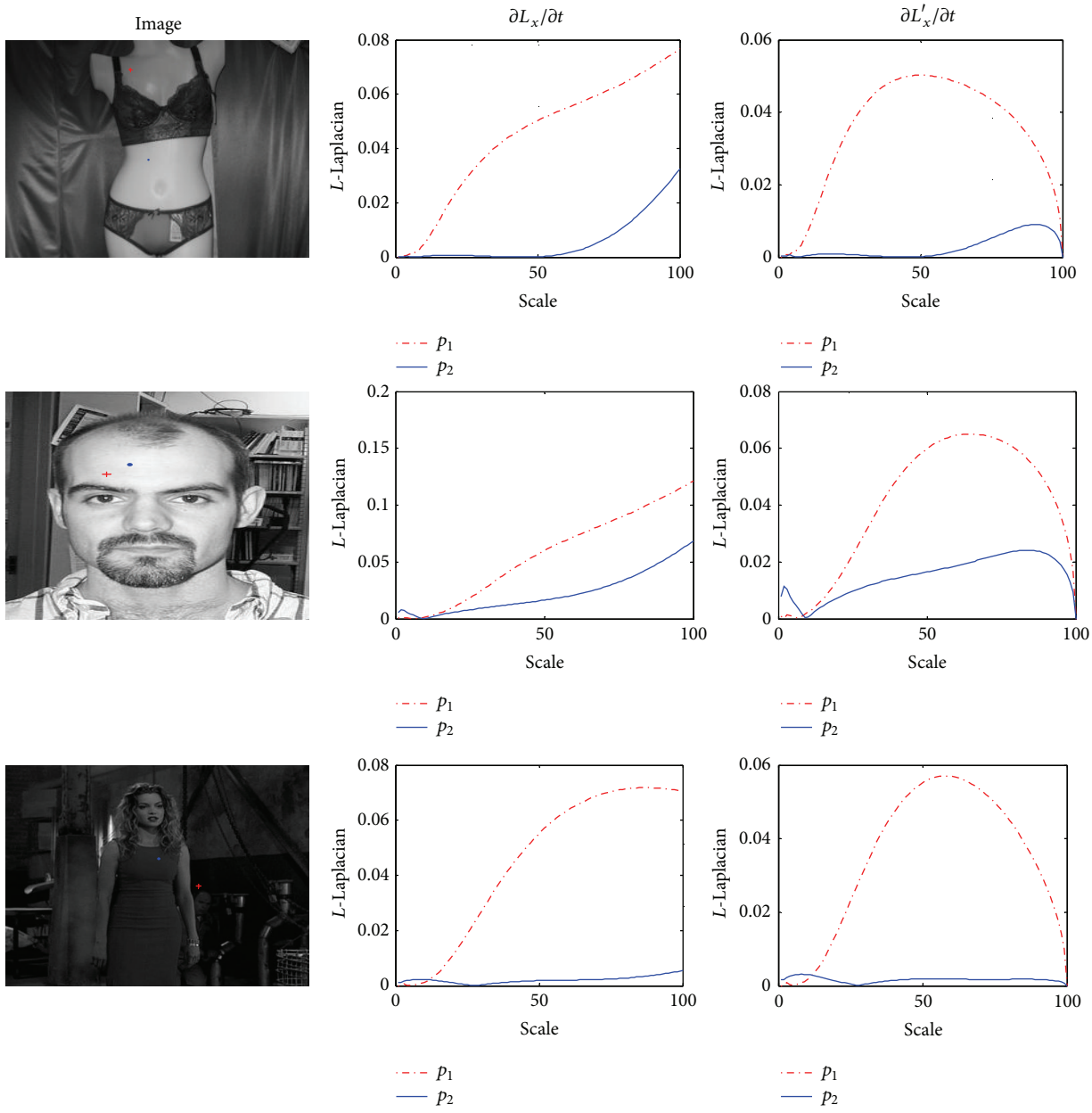


Figure 8: Results from improved texture measure method under different images.

determined, then according to property (ii), the threshold T_s can be defined as follows:

$$T_s = \min_i \left(\max_j \left\{ t\left(x_j\right) \mid x_j \in R_i \right\} \right), \quad \left\{ R_i \in I \mid N\left(R_i\right) \geq T_O \right\}, \tag{15}$$

where $t(x)$ is texture strength which is detected by the local maximum of the derivative over scales at point $x \in R_i$. Since the points in the weakly textured region should be retained as much as possible, we here use $\max_j\{t(x_j) \mid x_j \in R_i\}$ as the subthreshold detected in region R_i; on the other hand, there also exist points which are not weakly textured points in some parts of the weakly textured region, so for removing these points the global threshold here can be obtained as $\min_i(\max_j\{t(x_j) \mid x_j \in R_i\})$.

3.4. Composed Algorithm. The mechanisms for weakly textured point detection and fine positioning were proposed, respectively. And they can be used in various ways in point detection and point removing. For the purpose of experiments and validation, we have here integrated these two modules into a composed algorithm, which can be expressed in the following four-step procedure.

Stage 1. Given a constant integer M, which is the number of points needed to be detected by WTD, M can be adjusted according to the size of image. Formulas (2) and (3) are then

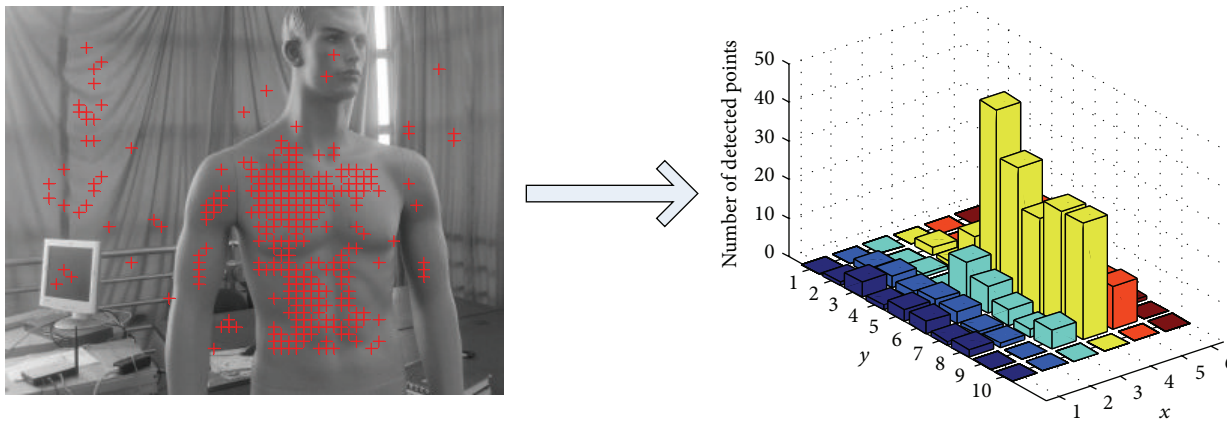


FIGURE 9: 3D histogram of weakly textured points distribution.

used to compute the vector in the radial-space $\{r_1, r_2, \ldots, r_N\}$ for each point p_i of image:

$$v_S(p_i) = \{S_{w\text{-texture}}(p_i, r_1), S_{w\text{-texture}}(p_i, r_2), \ldots, S_{w\text{-texture}}(p_i, r_N)\}. \quad (16)$$

Stage 2. Use the mechanism proposed in Section 2 to select the suitable radial r_{fit} for each point p_i of image I. Take $S_{w\text{-texture}}(p_i, r_{\text{fit}})$ as the similarity strength of p_i. And arrange p_i of image I in order of decreasing $S_{w\text{-texture}}(p_i, r_{\text{fit}})$. Then choose the first M points as the candidates of the weakly textured points.

Stage 3. Given a constant integer T, then scale t varies within $[1, 2, \ldots, T]$. Use $L'_x = p(t) \cdot L_x$ defined in formula (12) for computing the vector of each point which is obtained from Stage 2:

$$v_t(p_i) = \{L_{\text{grad}}(p_i, t_1), L_{\text{grad}}(p_i, t_2), \ldots, L_{\text{grad}}(p_i, t_T)\}, \quad t_i \in [1, 2, \ldots, T]. \quad (17)$$

Choose the biggest element from $v_t(p_i)$ as the texture strength of point p_i; that is,

$$v_{t\_\max}(p_i) = \max_{t_i \in [1,2,\ldots,T]} L_{\text{grad}}(p_i, t_i). \quad (18)$$

Stage 4. Using the threshold selection mechanism to detect the threshold T_s of texture strength and remove the points by T_s, that is, if $v_{t\_\max}(p_i) < T_s$, then p_i is the weakly textured point.

Figure 10 shows the performance of above composed algorithm; it is clear that this algorithm can efficiently extract the weakly textured points, and through the scale-space texture strength concept, the points have strong textureness which can be removed completely.

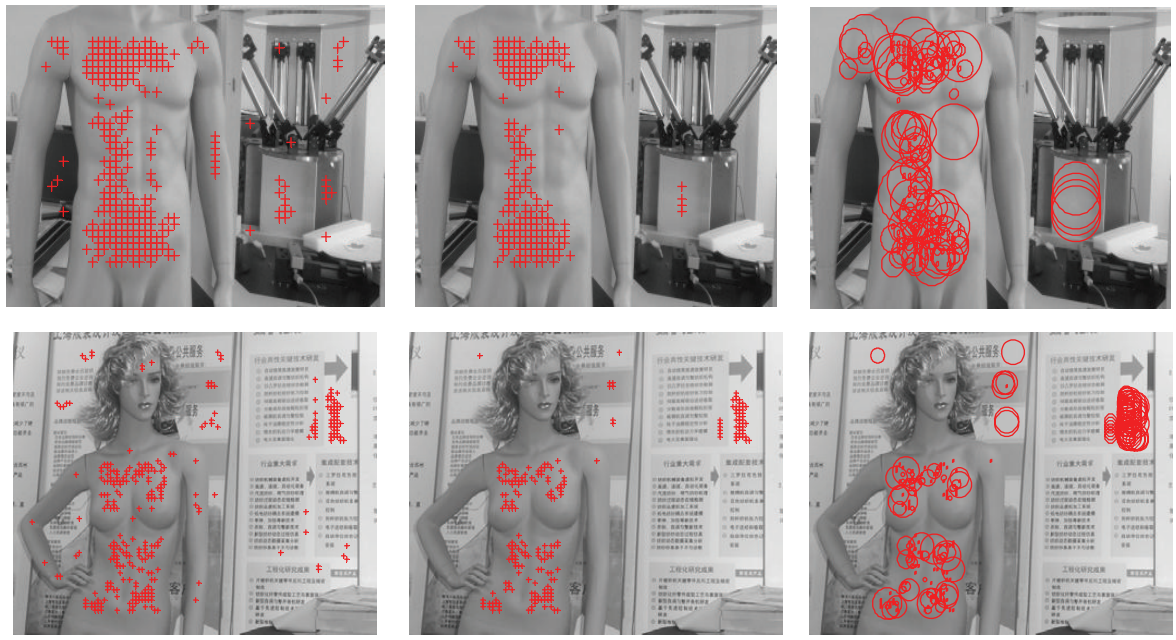
4. Sparse Representation for Point Correspondence

In this section, we study the feature matching problem of the weakly textured points detected by the proposed mechanism in the above two sections. Since the resulting feature matches can be used in wide-baseline stereo matching problem, it is important to find a matching algorithm for weakly textured points correspondence. On the other hand, since the point has the feature of weak texture, it is difficult to choose features for distinguishing different weakly textured points. Traditionally, methods of point-matching are based on the Euclidean distance; namely, if the distance between the candidate point and the object point is minimum, then those points are correspondence; moreover, the algorithms such as support vector machine (SVM) [25] and nearest neighbor algorithm [26] are largely dependent on the choice of features. Hence, we here match the feature via sparse representation, within the proposed framework; the precise choice of feature space is no longer critical. In [19], the algorithm used for human face recognition is similar to our method. And the efficacy of sparse representation in solving classification problems has been demonstrated.

4.1. Analyze the Feature of Weakly Textured Points. Since features of the same point in two different images are almost equal, the feature of the point in the first image can be thought as lying on the linear subspace composed by all features of weakly textured points in the second image. As proposed in [19], we here use the same hypothesis; that is, the feature from one point lies on a subspace. This is the only prior knowledge about our feature matching algorithm.

Given the feature set of points $S_2 = [s_{2,1}, s_{2,2}, \ldots, s_{2,n_2}] \in \Re^{m \times n_2}$ in the second image, where $s_{2,j}$ denotes feature of point p_j, if p_j exists in the first image, then its feature $s_{1,j}$ will be approximately in the linear span of S_2 associated with object j:

$$s_{1,j} = \alpha_1 s_{2,1} + \alpha_2 s_{2,2} + \cdots + \alpha_{n_2} s_{2,n_2}, \quad (19)$$



(a) (b) (c)

FIGURE 10: Results from threshold mechanism for different images. (a) Points detected by WTD. (b) Points detected by WTD + threshold mechanism. (c) Regions belong to detected points.

where $\alpha_i \in \Re$, $j = 1, 2, \ldots, n_2$. The linear representation $s_{1,j}$ also can be rewritten in terms of S_2 as

$$s_{1,j} = S_2 x \in \Re^m. \tag{20}$$

Since the point p_j existed in the two images, its feature in these two images should be closed enough. Hence, the coefficient vector x could be assumed as $x = [0, \ldots, 0, \alpha_j, 0, \ldots, 0]^T \in \Re^{n_2}$ where its entries are zero except those associated with jth point. And here we should solve the linear equation $s_{1,j} = S_2 x$; here $S_2 \in \Re^{m \times n_2}$, and n_2 is the number of the weakly textured points in the second image, and m is the dimension of the features. Here in order to reduce the computational complexity, the dimension of features should be chosen as small as possible; that is, it has $n_2 \gg m$, so the equation $s_{1,j} = S_2 x$ is underdetermined, and its solution is not unique. On the other hand, since the ideal solution x_0 of $s_{1,j} = S_2 x$ should be dense, as assumed above. So the denser x_0 is, the easier will it be to determine point p_j in the first image.

Through the above analysis, we here seek the sparsest solution to $s_{1,j} = S_2 x$, that is, solving the following optimization problem:

$$x_0 = \arg\min \|x\|_0 \quad \text{subject to } s_{1,j} = S_2 x, \tag{21}$$

where $\|\cdot\|_0$ denotes the ℓ_0-norm, which counts the number of nonzero entries in a vector. However, this problem is NP-hard and even difficult to approximate [27]: there are no more efficient procedures for finding the sparest solution than exhausting all subsets of the entries for x. But in the theory of sparse representation and compressed sensing [28], it is proved that if the solution of ℓ_0-minimization is sparse enough, then it is equal to the solution of the ℓ_1-minimization problem:

$$x_0 = \arg\min \|x\|_1 \quad \text{subject to } s_{1,j} = S_2 x. \tag{22}$$

We here use a simple example for interpretation of the reason why should we use the ℓ_1-minimization rather than ℓ_2-minimization to get the sparest solution. As shown in Figure 11, the left and right images show the geometry of ℓ_2-minimization algorithm and ℓ_1-minimization algorithm to solve $s_{1,j} = S_2 x_0$ under two-dimensional situation, respectively. According to Figure 11 the solution of ℓ_1-minimization is also the solution of ℓ_0-minimization; however, the one gotten from ℓ_2-minimization does not satisfy ℓ_0-minimization.

4.2. Feature Matching Based on Sparse Representation. So far, we have used the simple example for interpreting the sparse representation theory and before we give the algorithm, all the symbols used latter should be defined as follows.

I_1: the image having the points which should be matched.

I_2: the image having the reference points for those in image I_1.

$\{q_i\}_{I_1}, \{p_i\}_{I_2}$: the weakly textured points sets, which are detected in the I_1, I_2, respectively.

$F(\cdot) : \Re^2 \to \Re^m$: feature extraction function as the position of point; that is, it can extract the m-dimension feature for the point.

$\delta_i(\cdot)$: $\Re^n \to \Re^n$: the function that can get a new vector from the vector $x \in \Re^n$, and the new one

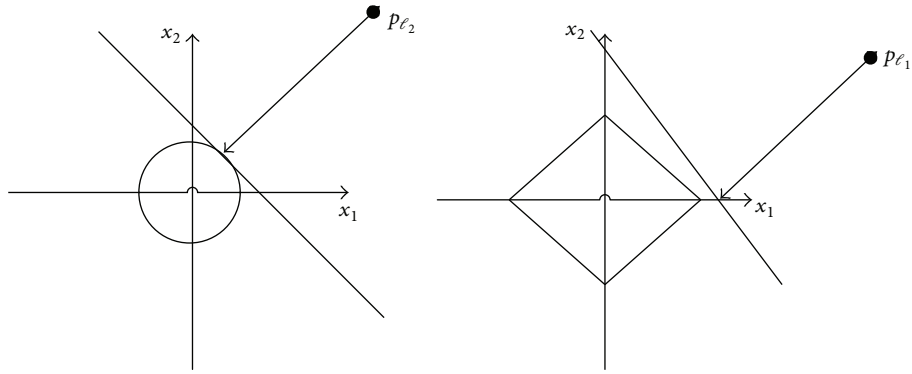


Figure 11: Geometry of sparse representation with ℓ_2-minimization and ℓ_1-minimization.

has the only nonzero entries in $x \in \Re^n$ which are associated with index i; moreover because the point in image I_1 can only have one corresponding point, so here the new vector gotten from $\delta_i(\cdot)$ should have only one nonzero entry.

Sparse Representation-Based Feature Matching Algorithm

(1) Input the feature set of reference points in image I_2. That is, $S_2 = \{F(p_1), F(p_2), \ldots, F(p_n)\} \in \Re^{m \times n_2}$ which is regarded as the matrix for ℓ_1-minimization problem. And the feature of test point $F(q)q \in I_1$, which can be regarded as a test sample.

(2) Normalize the columns of S_2 and $F(q)$ to have unit ℓ^2-norm; here $\overline{S}_2$ and $\overline{F}(q)$ are the normalized S_2 and $F(q)$, respectively.

(3) Solve the following ℓ_1-minimization problems:

$$x_0 = \arg\min \|x\|_1 \quad \text{subject to } \overline{F}(q) = \overline{S}_2 x \tag{23}$$

or

$$x_0 = \arg\min \|x\|_1 \quad \text{subject to } \left\|\overline{S}_2 x - \overline{F}(q)\right\|_2 \le \varepsilon, \tag{24}$$

where if $\overline{F}(q)$ have noise term, and the bounded energy of this noise is less than ε, then we will use the extended ℓ_1-minimization problem [19].

(4) Compute the residuals $r_i(\overline{F}(q)) = \|\overline{F}(q) - \overline{S}_2\delta_i(x_0)\|_2$ for $i = 1, \ldots, n$.

(5) Output identity $(\overline{F}(q)) = \arg\min_i r_i(\overline{F}(q))$, then the position of corresponding point can be determined by this result.

4.3. Feature Extraction for Weakly Textured Point. As above shown, we have proposed the feature matching algorithm for the weakly textured points based on the sparse representation classifier, but there still remains a problem, that is, what feature should be extracted. As assumed above the point feature in image I_1 should have the sparest solution in the feature space of image I_2; this means that if the feature satisfies this sparsity condition then it can be used for matching weakly textured points, that is, as in [19]: "the choice of features is no longer critical." On the other hand, this condition is satisfied by the feature of texture points, because these points have their own structure in their neighbor regions, and this feature cannot be represented by other points. However, the weakly textured points might not satisfy this condition, for the reason that these points usually have no obvious characteristics. So, in the experiment part we will use the different descriptors including LBP, random, and downsampled features to investigate their performance to the proposed matching algorithm. Here the random feature can be expressed as [29]

$$x_0 = \arg\min \|x\|_1 \quad \text{subject to } \widehat{y} = RSx, \tag{25}$$

where $R \in \Re^{d \times m}$ is a random matrix independently sampled from zero-mean normal distribution; moreover each row of R is normalized to unit length, and $\widehat{y} = Ry \in \Re^d$, when $d \ll m$; then using this feature can reduce the computation complex, and the polytope geometry as the analysis indicates the following: if the solution x_0 is sparse enough, then it can be correctly recovered by ℓ_1-minimization from any sufficiently large number d of linear measurements $\widehat{y} = RSx$ with the probability:

$$d \ge 2t \log\left(\frac{n}{d}\right), \tag{26}$$

where t is the number of nonzero entries in vector x.

For ending this section, we will give the method to decide the validation of the sample. In the real-world situation, when a point is detected in image I_1 might not be detected in image I_2 it means that the feature of this point could not be described by using only one point feature in image I_2; that is, the nonzero entries of its sparse representation in image I_2 are not concentrated on one subject; in contrary, it has widely spread sparse coefficients among multiple points in image I_2; hence, the sparsity concentration index (SCI) [19] to determine the validation of the sample is as follows:

$$\mathrm{SCI}(x) = \frac{k \cdot \max \left\|\delta_i(x)\right\|_1 / \|x\|_1 - 1}{k - 1} \in [0, 1]. \tag{27}$$

And according to this index then the valid point p can be decided if $\mathrm{SCI}(p) \ge \tau$, where $\tau \in (0, 1)$ is a threshold.

Figure 12: Examples of detected points distributed on different objects with different detectors.

Table 1: Performance under different objects.

Test images	Harris-affine		SIFT		SURF		WTD	
	OP/TP	OP	OP/TP	OP	OP/TP	OP	OP/TP	OP
Human face	0.1187	28.0000	0.05464	32.1429	0.1255	46.4286	0.5841	85.2857
Flower	0.1837	28.2857	0.0983	26.1428	0.2387	50.5714	0.7253	107.1429
Human model	0.1315	22.8571	0.0122	5.0000	0.1629	36.2857	0.8240	113.0000
Average	0.1446	26.38	0.05504	21.09523	0.1757	44.4285	0.7111	101.8095

5. Experiment Results

In this section, we present experiments on publicly available databases for weakly textured points detecting and the feature matching algorithm based on sparse representation, and the efficacy of these two algorithms can be demonstrated by these experiments. The experiments can be divided into three parts, that is, points detection, feature matching, and promoting the quasi-dense matching algorithm [20]; in the first part we will examine the performance of the detection algorithm, comparing with different images in various situations and comparing to several popular feature detectors (as shown in Sections 5.1 and 5.2). In the second part we will examine feature matching algorithm, comparing performance across various features spaces and feature dimensions and comparing to several popular classifiers in Section 5.3. Finally, we will demonstrate the proposed method for improving the quasi-dense matching algorithm in Section 5.4.

5.1. WTD with Image Perturbations. We here measure the WTD efficiency for weakly textured points detection under images with different objects and the robustness across with the same object has different backgrounds. The data we choose here include 200 images from Caltech human faces set, 102 images from category flower dataset, and 100 images from human model. And first we will compare our detector with three classical detectors in weakly textured points detection, namely, Harris-affine, SIFT, and SURF. For measuring the performance here we use the proportion between the number of detected points (Figure 12) on the objects (OP) and total number of detected points (TP) and set the radial of WTD from $R = 2$ to $R = 32$. Table 1 compares WTD to the other three detectors.

Table 2: Performance under different objects with different backgrounds.

Test images		Human face	Flower	Human body	Average
Harris-affine	RB	0.1242	0.0758	0.0380	0.0794
	FB	0.1111	0.0765	0.0234	0.0703
	TB	0.1228	0.0556	0.0208	0.0664
SIFT	RB	0.0778	0.0059	0.0009	0.0282
	FB	0.0281	0.002	0.0042	0.0114
	TB	0.4242	0.0136	0.0167	0.1516
SURF	RB	0.1925	0.1843	0.2135	0.1968
	FB	0.2131	0.1808	0.1566	0.1835
	TB	0.2157	0.1767	0.0948	0.1624
WTF	RB	0.8750	1	1	0.9583
	FB	0.9123	1	1	0.9708
	TB	0.8361	1	1	0.9454

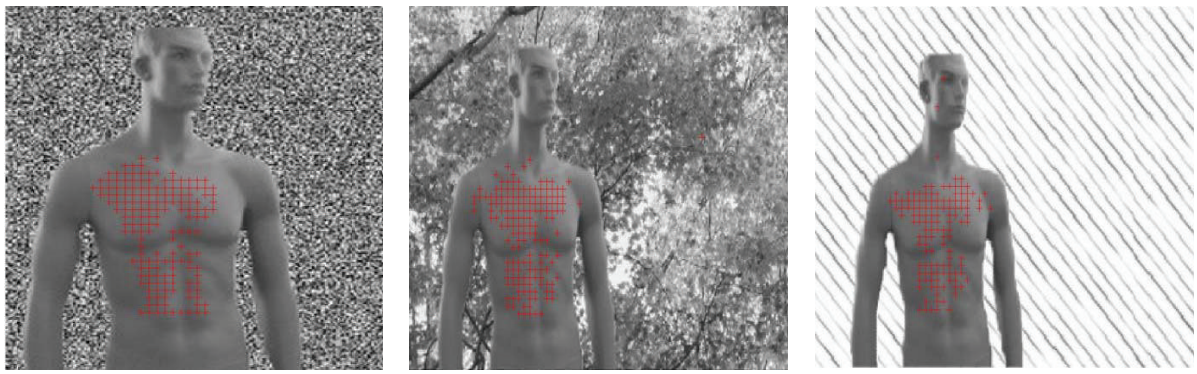


Figure 13: Performance of WTD on different backgrounds.

Based on the results on the data, we draw the following conclusion.

(1) For all data, the performances of WTD consistently exceed the other three detectors in weakly textured detection. It means that if the object has large region of weak texture, then WTD can detect more points in this object than other detectors.

(2) The results obtained by WTD and classical detectors are complementary; namely, the points which are not detected by classical detectors can be detected by WTD, so if points detected by those classical detectors are not enough for the stereo reconstruction, then WTD can be used as a complement.

As a second set of this experiment, the robustness of WTD for the same object with three different texture backgrounds is tested. In order to demonstrate the performance of WTD, we here use artificial backgrounds as the extreme situations, namely, random background (RB), forest background (FB), and texture background (TB), respectively. And the parameters of WTD are the same as in the above experiment. Table 2 and Figure 13 show the performance of WTD comparing to classical detectors.

It is clear that the points detected by WTD are almost concentrated on the object with weak texture (human model) no matter what texture background takes (the WTP/TP close to 1). On the other hand, this property of WTD is contrary to classical detectors, whose detected points are almost concentrated on the texture objects, and it is once again demonstrating the relationship between WTD and classical detectors as the above experiment.

5.2. Performance under Varying Blur, Lighting Change, Rotating, and Viewpoint Change. For this experiment, we test the repeatability of four detectors under different photometric and geometric transformations, as the protocol suggested in [13]. Test images sets and results are shown in Figures 14–17 and each set changes in 6 levels. Because of space constraints, we here only give three of the six images in each type of transformation, that is, first, third, and sixth. Each figure presents the repetition rate which includes three parts, namely, total number of matched points, the number of matched points on the objects, and the matching score on the objects. The range of radius scale is set from $R = 2$ to $R = 30$, and the repetition rate here we just use the region matches, and the matching conditions follow as recommended in [22]. The matching score S_i, here, is given by the proportion of

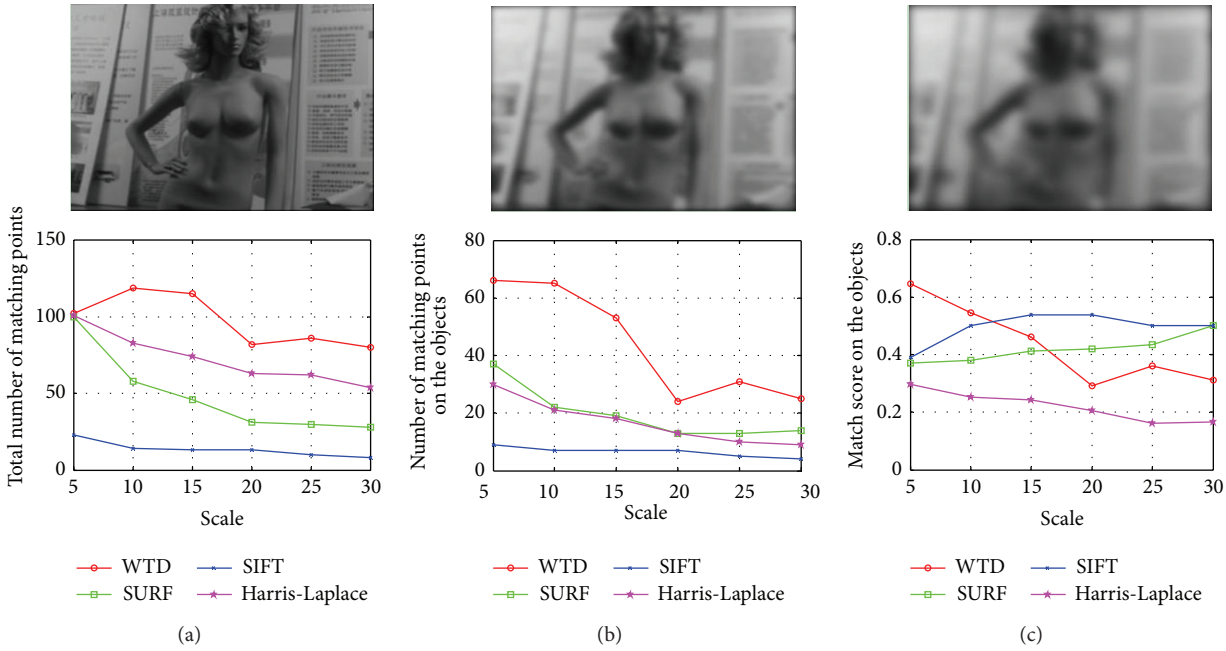


FIGURE 14: Blur for image. (a) Total number of matched points for image blurs. (b) Number of matched points on the objects. (c) Matching score on the objects.

correspondences with detected regions on the objects for all correspondences in the test image; that is,

$$S_i = \frac{\text{Total number of matches}}{\text{Total number of detected regions}} = \frac{N_i}{N}, \quad (i = 1, 2, \ldots, M). \tag{28}$$

5.2.1. Blur. Figure 14 shows the results detected by four detectors undergoing increasing amounts of image blur, and we here use the Gaussian kernel to blur the image, and the scale parameter t (scale selection) is set as $t = 0, 10, \ldots, 60$; according to the results of all detectors, the number of matching points declined with the blur increasing; on the one hand, the reason for classical detectors is the image texture reduced by image blur. On the other hand, the number of match points detectors by our detector should increase with the blur, but in fact the threshold selection mechanism discussed in the previous section would get the small threshold because the blur causes the number of candidate regions increasing, so according to formula (15), the threshold will become smaller than before; hence the number of points detected by WTD will decline.

5.2.2. Lighting Changes. Figure 15 shows the results for lighting changes with four different detectors. From this result we can find the following: (a) the total number of match points gotten by WTD is lowest and (b) the match points on the object (human face) are higher than other detectors. This is caused by following reasons.

(1) The background of image chosen by us is clatter, so the total number of match points chosen by classical detectors is higher than WTD.

(2) When the intensity decreases, then the texture of background will increase, and the texture in the object is also increased with light intensity decreasing, but it is slower than the increase in the background; hence, the number of match points on the object by WTD is staying close to its mean value.

(3) The points detected by WTD are almost concentrated on the object, so its match score is large than others.

5.2.3. Rotation. Figure 16 shows the performance for image rotation; the number of match points on the object detected by WTD is not stable with the image rotation. This is because the object scale here will vary with the rotating process. And the same radial (see Section 2) used to measure the same weakly textured point in these images will get the different values. On the other hand, the backgrounds of these images are also textureless, and if the scale changed the threshold for reducing the texture points will also change; hence, points detected by WTD will concentrate on the background. Although being in such serious situation, the match score of WTD is also higher than other detectors.

5.2.4. Viewpoint Changes. Figure 17 shows the performance for viewpoint changes. It is clear that when viewpoint changes the matching score of WTD is always close to 1. This implicates that the matching points gotten from WTD are almost concentrating on the object with the viewpoint changes in 15° degree limitation.

5.3. Feature Matching Algorithm Experiment. In this part, we will test the performance of matching algorithm based on sparse representation with the detected weakly textured

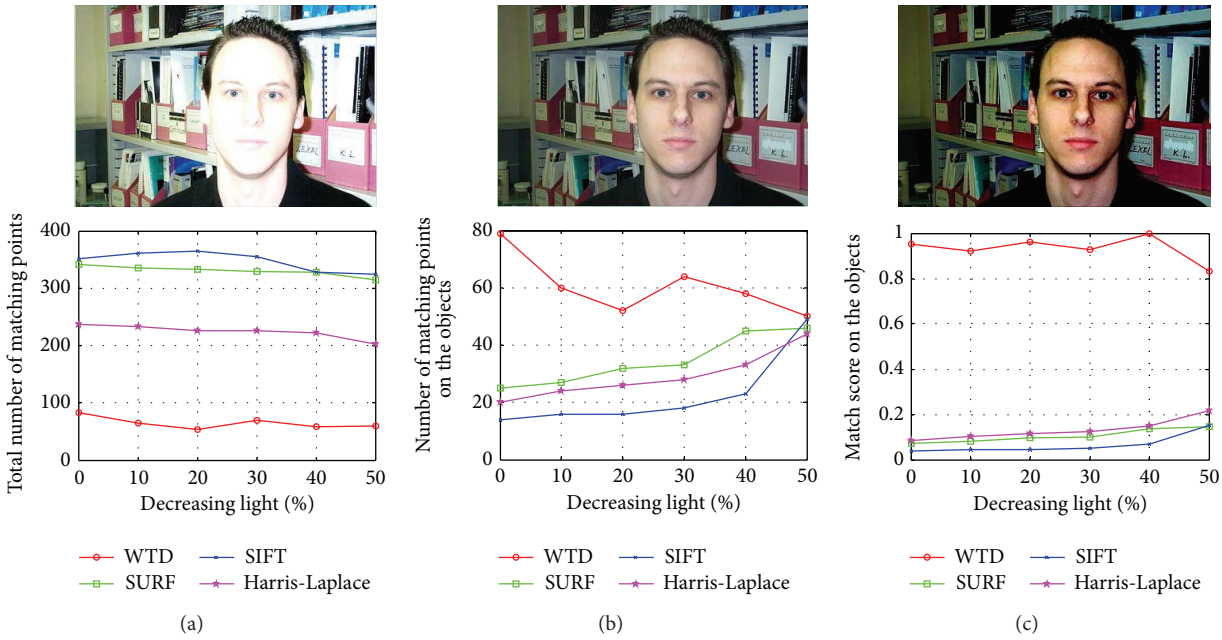


FIGURE 15: Lighting changes for image. (a) Total number of matched points for lighting change. (b) Number of matched points on the objects. (c) Matching score on the objects.

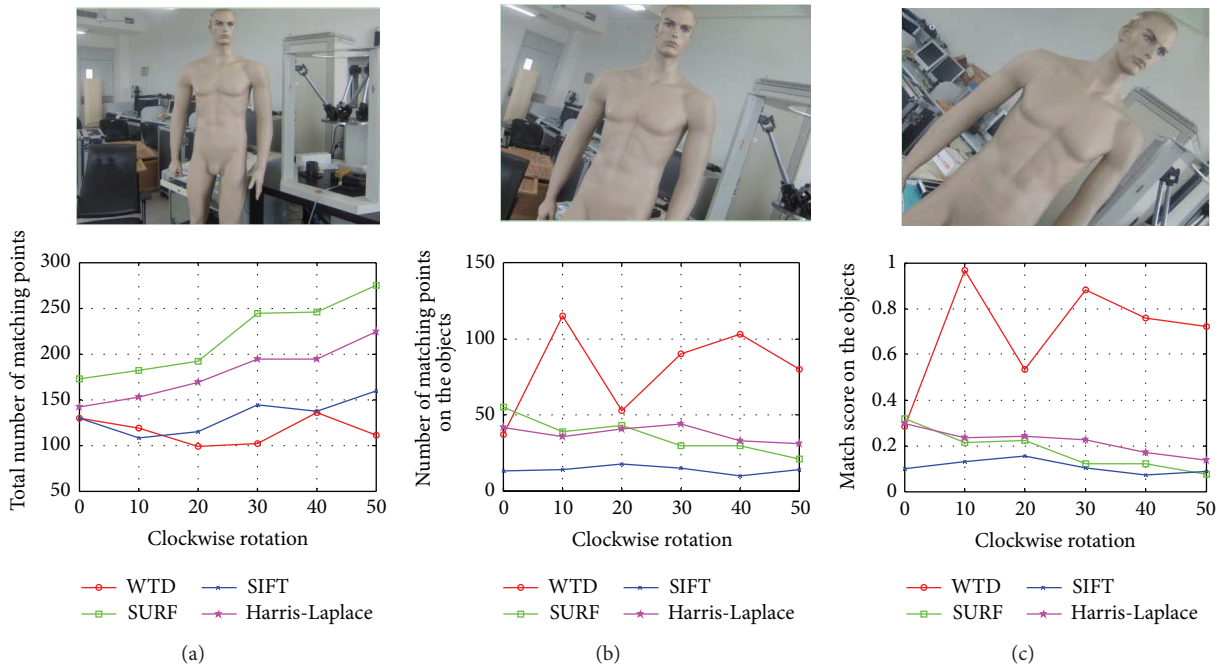


FIGURE 16: Rotation for image. (a) Total number of matched points for rotation. (b) Number of matched points on the objects. (c) Matching score on the objects.

points. We here compare performance across various feature spaces and feature dimensions with two popular classifiers, namely, linear SVM and ℓ^2-min. Moreover, we will also test our matching algorithm under random pixel corruption and random block occlusion, respectively.

5.3.1. Sparse Representation Based Weakly Textured Point Matching. We match 150 weakly textured points detected by WTD in each image as shown in Figure 18. Here we only test the matching algorithm under the WTD, and we use conventional features LBP (local binary pattern) and two unconventional features: random and downsampled region features. The window size of LBP is set to be 30×30 pixels, and we compute the matching rate with the feature space dimensions $10, 20, \ldots, 60$. The dimension of random feature space discussed in the previous section is set the same as

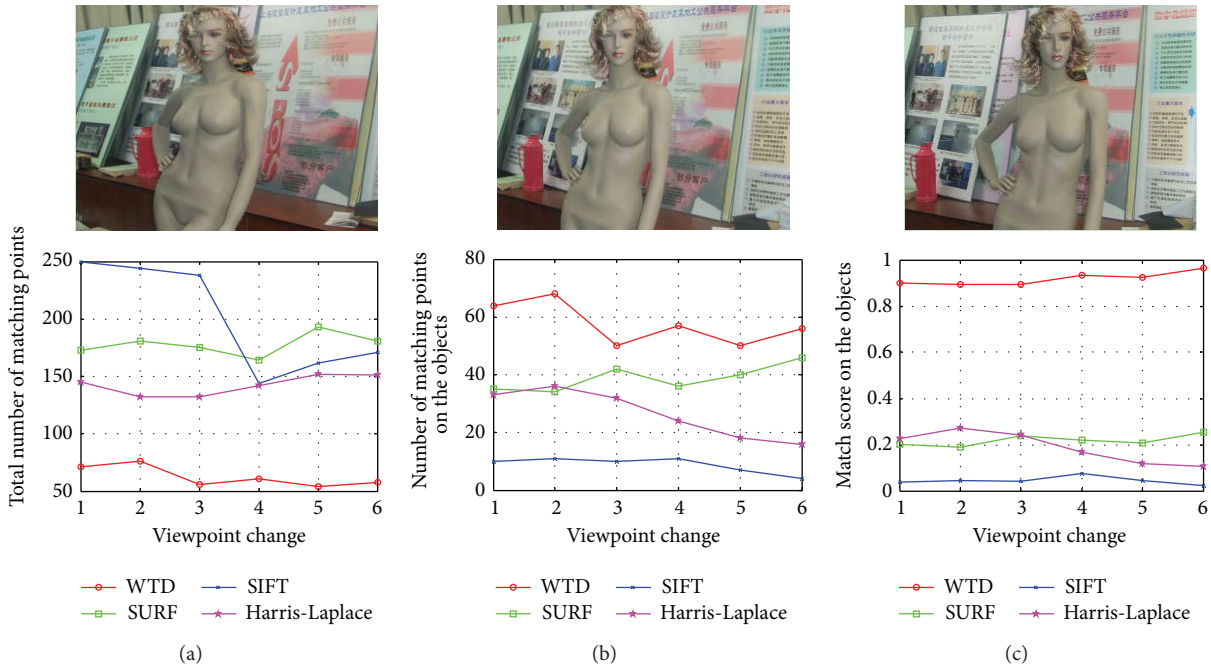


Figure 17: Viewpoint change for image. (a) Total number of matched points for viewpoint change. (b) Number of matched points on the objects. (c) Matching score on the objects.

Table 3: Variances of matching rate.

Algorithm	LBP	Downsample	Random
SRM	1.15×10^{-2}	9.8×10^{-3}	7.9540×10^{-4}
ℓ^2-min	2.47×10^{-2}	7.2×10^{-3}	5.6×10^{-3}
SVM	2.63×10^{-2}	4.29×10^{-2}	2.63×10^{-2}

LBP, namely, the random matrix $R^{d\times 60}$ $(d = 10, 20, \ldots, 60)$; finally the window size of downsampled feature is set to be 30×30 with dimensions 60, 45, 30, and 12. Those numbers correspond to downsampling ratios of 1/15, 1/20, 1/30, and 1/75, respectively. Figure 18 shows the match performance for the various features, in conjunction with three different algorithms: SRM, ℓ^2-min, and SVM.

Figure 18 also shows results under LBP descriptor with 40 dimensions. The algorithm returns enough matched points on the low texture region that lets us get more sufficient points than traditional sparse matching on these area.

Based on the results on different feature spaces, the following conclusion can be drawn.

(1) Here we use the variance of feature dimensions to measure the stability of matching rate on different feature spaces. Table 3 shows the variances of matching rate for SRM, ℓ^2-min, and SVM. It is clear that SRM is more stable than others for feature dimensions on the LBP and random spaces.

(2) The biggest matching rate of SRM exceeds the best performances of others on the LBP and downsample spaces. More specifically, the best performance for SRM on the LBP is 80.86%, compared to 71.35% for ℓ^2-min and 69.70% for SVM. The best rate for SRM on the downsample is 80.38%, compared to 59.64% for ℓ^2-min and 55.97% for SVM.

(3) The results on the LBP and random spaces suggest that when the feature dimension is 40, it is sufficed for sparse recovery. Moreover when the dimensions are beyond 40, the performances on these two feature spaces will converge.

5.3.2. Matching despite Random Pixel Corruption. In this experiment, we test the robustness of SRM under the pixel of description region occluded by random noise. We use the extended ℓ_1-minimization problem [19] at the third step of SRM for solving this occlusion problem. Here the error tolerance ε is set to be the bounded energy of the random noise. In order to eliminate the influence caused by WTD, we use the random noise to corrupt the located region in which the feature is extracted (see Figure 19). On the other hand, since there are no efficient descriptions for weakly textured point, the description we used is LBP, because it has the best performance as shown above. For demonstrating the performance of SRM, we compare it to the following algorithms: $\ell^2_{\min}$ + PCA, $\ell^2_{\min}$ + LNMF, and $\ell^2_{\min}$ + ICA, respectively. Here PCA, LNMP, and ICA are principal component analysis, local nonnegative matrix factorization (LNMF), and independent component analysis (ICA), respectively. These algorithm are used for feature preprocessing and the ℓ^2-min we used here because it has the second highest matching rate in the above experiment. According to the experiment result, it can be seen that when the corruption is up to 25 percent the number of matching points with each algorithm will drop to zero. So, here we only show the result with corruption from 0 percent to 25 percent. According to the performance of

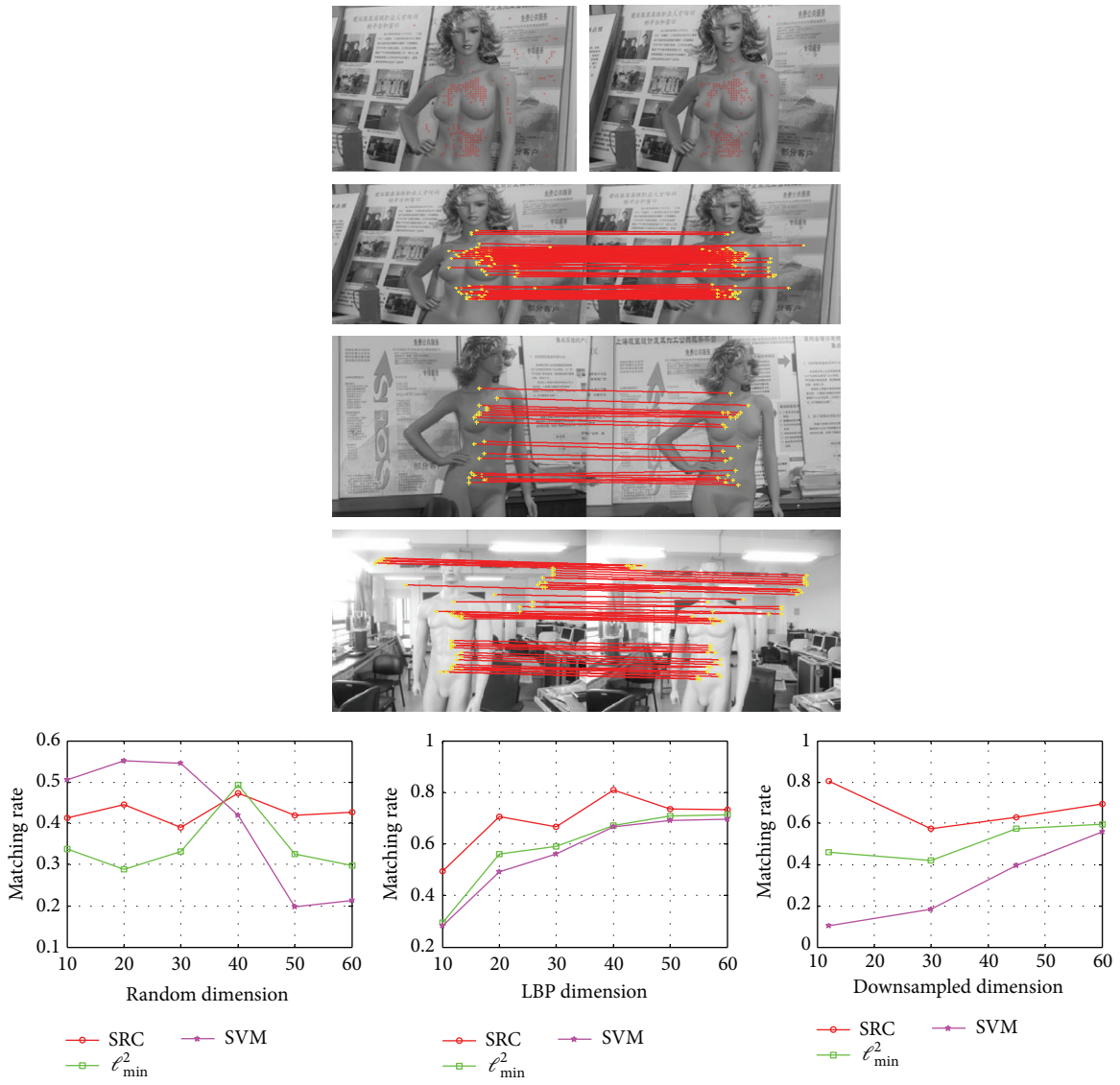


Figure 18: Matching rate for various feature transformations.

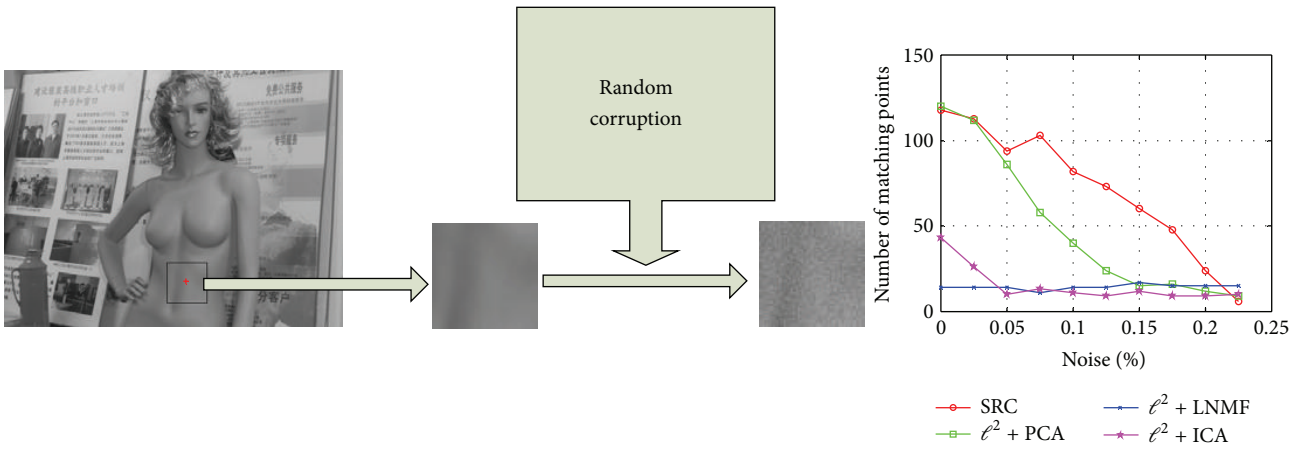


Figure 19: Recognition under random corruption.

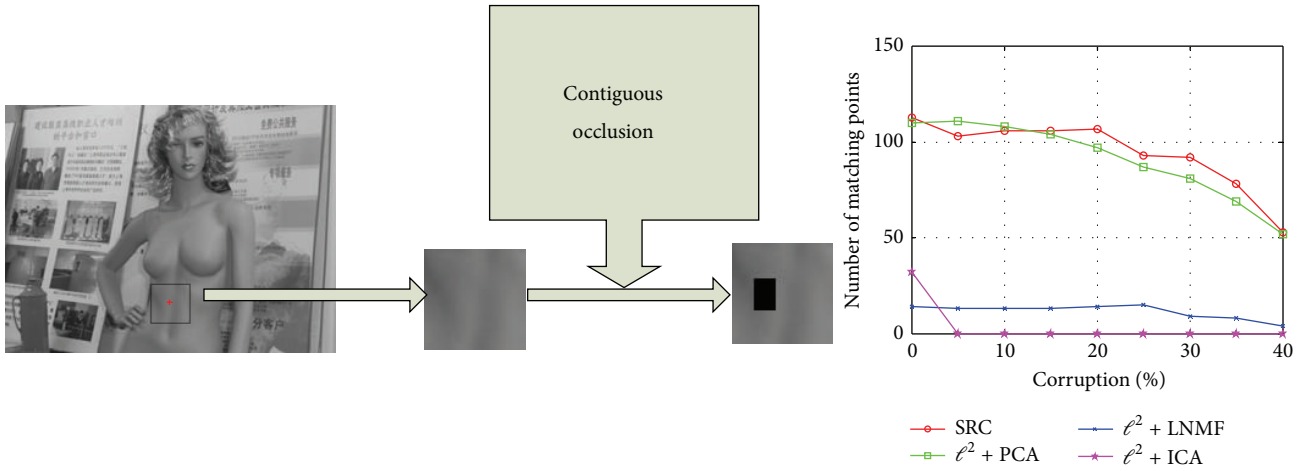


Figure 20: Recognition under varying level of contiguous occlusion.

SRM and its three competitors, we see that SRM outperforms others; namely, from 5 percent up to 20 percent occlusion, the number of matching weakly textured points are higher than others. At 10 percent corruption, the highest matching number of others is 40 points, while the one gotten by SRM 80 points. Even at 15 percent occlusion, the matching number is still 60 points. Clearly, the SRM can ensure to tolerate under the corruption less than 25 percent for matching the weakly textured points. On the other hand, the matching points numbers gotten by $\ell^2_{\min}$ + LNMF and $\ell^2_{\min}$ + ICA are obviously less than SRM and $\ell^2_{\min}$ + PCA; it has been suggested that both LNMF and ICA are not suitable for the preprocessing of weakly textured feature.

5.3.3. Matching despite Random Block Occlusion. The final part of the expe riment here is to simulate various levels of contiguous occlusion, from 0 percent to 40 percent, by replacing a randomly located square block of each feature region with the gray-lever at zero, and the reason why we choose zero gray-lever block is that the matching points are weakly textured point, so if we choose the texture block here, the characteristic of these points will be replaced by the features in this texture block. And the region here is the same as the discussion in the above experiment (see Figure 20). The location of occlusion is randomly chosen for each image and is unknown to the computer. The results got from SRM and $\ell^2_{\min}$ + PCA are still better than others. Again, the above implication is also true for this case. Despite the fact that the performances of SRM and $\ell^2_{\min}$ + PCA are closed, it is clear that SRM outperforms $\ell^2_{\min}$ + PCA. For example, at 20 percent occlusion, SRM achieves 109 points; it is higher than $\ell^2_{\min}$ + PCA's 98 points. On the other hand, the reason why the number of matching points is not decreased to zero when occlusion increases to 40 percent is that the located square block we used here is the zero gray-lever images. And the matching points also have weakly textured characteristic, so it will not be much affected by this square block.

5.4. Improved Quasi-Dense Matching. In this section, we use WTD to promote the quasi-dense matching algorithm [20]. The quasi-dense matching algorithm starts from a set of sparse seed matches which were usually obtained by classical detectors, then propagates to the neighboring pixels by the best-first strategy, and finally produces a quasi-dense disparity map. Since most initial sparse seed matches distribute in the strong texture regions, it nearly has no seed matches in the textureless areas; furthermore if the matches got from propagating step in these areas are wrong, gross reconstruction errors will occur, so here we use the proposed approach to get the initial sparse seed matches in the large textureless areas.

Table 4: Comparison between original dense matching and improved one under different circumstances.

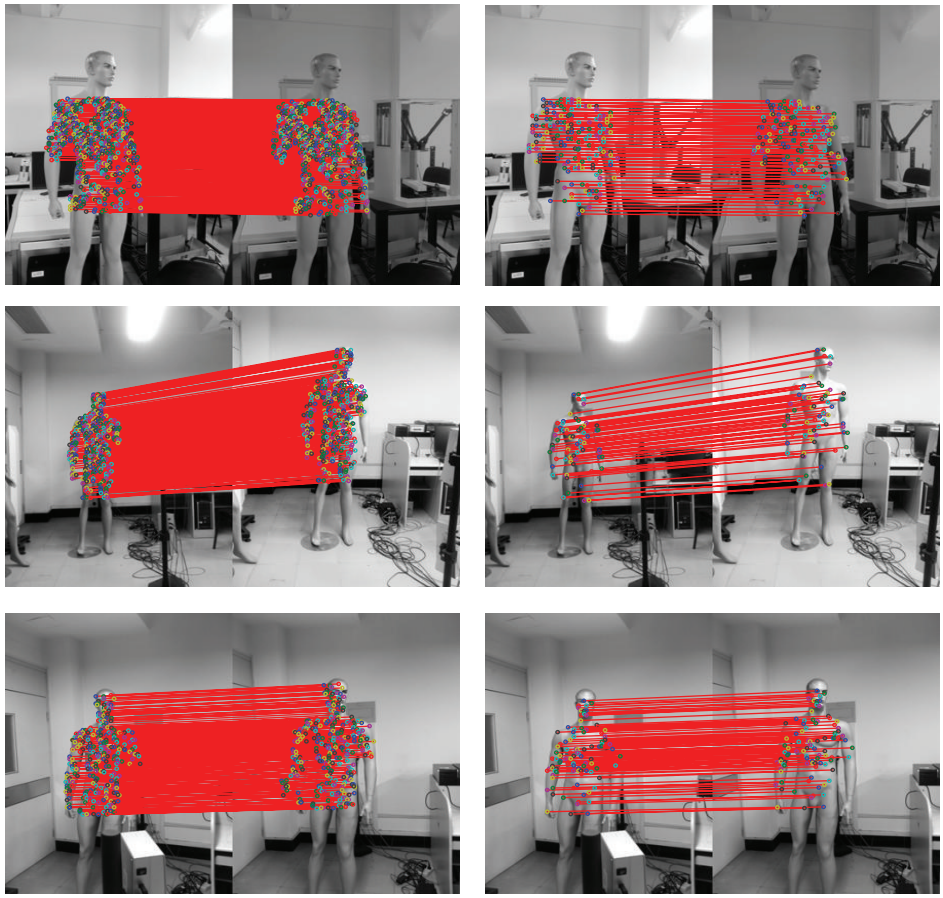
Change	Algorithm	#corresponding-point on untex
Lighting	Quasi + ours	845
	Quasi	414
Image scale	Quasi + ours	490
	Quasi	210
Camera angle	Quasi + ours	676
	Quasi	351

The original algorithm is independently implemented for comparison on the same image pairs. The corresponding-point sets obtained by WTD and traditional quasi-dense matching algorithm are shown in Figure 21. One can verify the effectiveness of our implementation by corresponding quality measures which are given in Table 4.

Our algorithm is evaluated on the changes of lighting, scale, and camera angle, respectively, and the results demonstrate that the algorithm performs particularly better on the large textureless object surface than the original one (as shown in Figure 21).

6. Conclusion

In this paper, we proposed an efficient detector for weakly textured points and used sparse representation for matching the detected weakly textured points in two different images.



(a) (b)

Figure 21: Results from improved dense matching and original one under different circumstances. Top row: corresponding-points matched with lighting changes. Middle row: corresponding-points matched with scale changes. Bottom row: corresponding-points matched with camera angle changes. (a) Corresponding results from improved quasi-dense matching algorithm. (b) Corresponding results from original quasi-dense matching algorithm.

We have contended both theoretically and experimentally that the proposed algorithms outperform others in weakly textured points detecting and matching. The performance of the new detector was demonstrated on a wide variety of image sets in which the weakly textured objects were included. The proposed detector gives a way of detecting the weakly textured points which would be useful for 3D reconstruction of the object with weak texture at the dense matching step. On the other hand, the proposed matching algorithm SRM has the higher performance than others on the LBP feature space; moreover, to a certain extent, SRM can handle occlusion and corruption on the feature of weakly textured points.

Intriguing questions for future work are whether this method is appropriate for wide-based stereo and what description can be used for these weakly textured points. The first problem can be solved by replacing circumferences with ellipses; the second one might be solved by using the information of the surround points which are detected by classical detectors, such as SIFT, GOLH, and SURF, and it is a most challenging one as well. These are two directions for our future work.

Conflict of Interests

The authors declare that there is no conflict of interests regarding the publication of this paper.

Acknowledgments

This work was supported in part by the Key Project of the National Nature Science Foundation of China (no. 61134009), the National Nature Science Foundation of China (no. 61473078), Cooperative research funds of the National Natural Science Funds Overseas and Hong Kong and Macao scholars (no. 61428302), Specialized Research Fund for Shanghai Leading Talents, Project of the Shanghai Committee of Science and Technology (nos. 13JC1407500, 11JC1400200), and Innovation Program of Shanghai Municipal Education Commission (no. 14ZZ067).

References

[1] Y. Boykov, O. Veksler, and R. Zabih, "Fast approximate energy minimization via graph cuts," *IEEE Transactions on Pattern*

Analysis and Machine Intelligence, vol. 23, no. 11, pp. 1222–1239, 2001.

[2] Y. Boykov and V. Kolmogorov, "An experimental comparison of min-cut/max-flow algorithms for energy minimization in vision," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 26, no. 9, pp. 1124–1137, 2004.

[3] C. Harris and M. Stephens, "A combined corner and edge detector," in *Proceedings of the Alvey Vision Conference*, pp. 147–151, 1988.

[4] K. Mikolajczyk and C. Schmid, "Indexing based on scale invariant interest points," in *Proceedings of the 8th International Conference on Computer Vision (ICCV '01)*, vol. 1, pp. 525–531, Vancouver, Canada, July 2001.

[5] T. Lindeberg, "Feature detection with automatic scale selection," *International Journal of Computer Vision*, vol. 30, no. 2, pp. 79–116, 1998.

[6] D. G. Lowe, "Distinctive image features from scale-invariant keypoints," *International Journal of Computer Vision*, vol. 60, no. 2, pp. 91–110, 2004.

[7] K. Mikolajczyk and C. Schmid, "An affine invariant interest point detector," in *Computer Vision—ECCV 2002*, vol. 2350 of *Lecture Notes in Computer Science*, pp. 128–142, Springer, Berlin, Germany, 2002.

[8] F. Jurie and C. Schmid, "Scale-invariant shape features for recognition of object categories," *Computer Vision and Pattern Recognition*, vol. 2, pp. 90–96, 2004.

[9] T. Tuytelaars and L. van Gool, "Wide baseline stereo matching based on local, affinely invariant regions," in *Proceedings of the British Machine Vision Conference (BMVC '00)*, pp. 412–422, 2000.

[10] H. Bay, T. Tuytelaars, and L. van Gool, "SURF: speeded up robust features," in *Computer Vision—ECCV 2006*, vol. 3951 of *Lecture Notes in Computer Science*, pp. 404–417, Springer, Berlin, Germany, 2006.

[11] M. J. Atallah, "On symmetry detection," *IEEE Transactions on Computers*, vol. 34, no. 7, pp. 663–666, 1985.

[12] G. Loy and A. Zelinsky, "Fast radial symmetry for detecting points of interest," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 25, no. 8, pp. 959–973, 2003.

[13] J. Maver, "Self-similarity and points of interest," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 32, no. 7, pp. 1211–1226, 2010.

[14] V. G. Kim, Y. Lipman, and T. Funkhouser, "Symmetry-guided texture synthesis and manipulation," *ACM Transactions on Graphics*, vol. 31, no. 3, article 22, 2012.

[15] T. Lindeberg, *Scale-Space Theory in Computer Vision*, Springer, Boston, Mass, USA, 1994.

[16] R. Deriche, Z. Zhang, Q. Luong, and O. Faugeras, "Robust recovery of the epipolar geometry for an uncalibrated stereo rig," in *Computer Vision—ECCV '94*, pp. 567–576, 1994.

[17] D. Aiger and K. Kedem, "Exact and approximate geometric pattern matching for point sets in the plane under similarity transformations," in *Proceedings of the 19th Canadian Conference on Computational Geometry (CCCG '07)*, pp. 181–184, Ottawa, Canada, August 2007.

[18] D. Donoho, H. Kakavand, and J. Mammen, "The simplest solution to an underdetermined system of linear equations," in *Proceedings of the IEEE International Symposium on Information Theory (ISIT '06)*, pp. 1924–1928, Seattle, Wash, USA, July 2006.

[19] J. Wright, A. Y. Yang, A. Ganesh, S. S. Sastry, and Y. Ma, "Robust face recognition via sparse representation," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 31, no. 2, pp. 210–227, 2009.

[20] M. Lhuillier and L. Quan, "A quasi-dense approach to surface reconstruction from uncalibrated images," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 27, no. 3, pp. 418–433, 2005.

[21] K. Mikolajczyk, T. Tuytelaars, C. Schmid et al., "A comparison of affine region detectors," *International Journal of Computer Vision*, vol. 65, no. 1-2, pp. 43–72, 2005.

[22] T. Kadir, A. Zisserman, and M. Brady, "An affine invariant salient region detector," in *Computer Vision-ECCV 2004*, pp. 345–457, 2004.

[23] C. Sun, "Fast stereo matching using rectangular subregioning and 3D maximum-surface techniques," *International Journal of Computer Vision*, vol. 47, no. 1–3, pp. 99–117, 2002.

[24] P. Perona and J. Malik, "Scale-space and edge detection using anisotropic diffusion," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 12, no. 7, pp. 629–639, 1990.

[25] C. Wallraven, B. Caputo, and A. Graf, "Recognition with local features: the kernel recipe," in *Proceedings of the 9th IEEE International Conference on Computer Vision*, pp. 257–264, Nice, France, October 2003.

[26] S. Cost and S. Salzberg, "A weighted nearest neighbor algorithm for learning with symbolic features," *Machine Learning*, vol. 10, no. 1, pp. 57–78, 1993.

[27] E. Amaldi and V. Kann, "On the approximability of minimizing nonzero variables or unsatisfied relations in linear systems," *Theoretical Computer Science*, vol. 209, no. 1-2, pp. 237–260, 1998.

[28] E. J. Candes and T. Tao, "Near-optimal signal recovery from random projections: universal encoding strategies?" *IEEE Transactions on Information Theory*, vol. 52, no. 12, pp. 5406–5425, 2006.

[29] C. Liu, "Capitalize on dimensionality increasing techniques for improving face recognition grand challenge performance," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 28, no. 5, pp. 725–737, 2006.

Hindawi Publishing Corporation
Mathematical Problems in Engineering
Volume 2014, Article ID 393265, 12 pages
http://dx.doi.org/10.1155/2014/393265

Research Article

An MPCA/LDA Based Dimensionality Reduction Algorithm for Face Recognition

Jun Huang,[1] Kehua Su,[2] Jamal El-Den,[3] Tao Hu,[1] and Junlong Li[2]

[1] *The State Key Laboratory of Information Engineering in Surveying, Mapping and Remote Sensing, Wuhan University, Wuhan 430072, China*
[2] *School of Computer, Wuhan University, Wuhan 430072, China*
[3] *School of Engineering and IT, Charles Darwin University, Darwin, NT 0909, Australia*

Correspondence should be addressed to Kehua Su; skhemail@163.com

Received 10 January 2014; Revised 17 July 2014; Accepted 23 July 2014; Published 31 August 2014

Academic Editor: Yi Chen

We proposed a face recognition algorithm based on both the multilinear principal component analysis (MPCA) and linear discriminant analysis (LDA). Compared with current traditional existing face recognition methods, our approach treats face images as multidimensional tensor in order to find the optimal tensor subspace for accomplishing dimension reduction. The LDA is used to project samples to a new discriminant feature space, while the K nearest neighbor (KNN) is adopted for sample set classification. The results of our study and the developed algorithm are validated with face databases ORL, FERET, and YALE and compared with PCA, MPCA, and PCA + LDA methods, which demonstrates an improvement in face recognition accuracy.

1. Introduction

Face recognition has become a topical and timely study focus in the fields of pattern recognition and computer vision for its wide application prospect [1, 2]. Feature extraction is the key element in face recognition. Currently, diverse recognition methods use different extraction strategies. And one of the most popular algorithms is principal component analysis algorithm (PCA), which aims to find the projected directions along with the minimum reconstructing error and then map the face dataset to a low-dimensional space spanned by those directions corresponding to the top eigenvalues [3, 4]. Traditional PCA face recognition technology can reach accuracy rate of 70%–92% [5]. However, it is still not fully practical.

PCA has certain limitations which result in bad adaptability in the image brightness and facial expression variety [6–9]. Under either strong bright light or weak light environments, the information of the features of the face is deficient; hence the structural information from the feature points of the face image may hardly be captured using traditional algorithms like PCA [10]. In addition, existing algorithms which are based on capturing single expressions make it difficult and challenging to capture the correct features of the same person if he changes his facial expressions. Traditional PCA fails to see the natural structure and correlation represented in data set [3], which leads to potential additional loss of compact and/or useful facial representations and will result in a higher reconstruction error rate [11].

There are many recognition proposals to address limitations of PCA presented above. In [12], Bansal and Chawla proposed normalized principal component analysis (NPCA) to improve the recognition rate. They normalized images to remove the lightening variations by applying SVD instead of eigenvalue decomposition. Pereira et al. [13] introduced a new technique which can reduce face dimensions called class-modular image principal component analysis (CMIPCA) to extract local and global information to reduce illumination effects, face expressions, and head-pos changes resulting in speed-up over PCA. In [14] Tsai showed an application of dimensionality reduction techniques, such as PCA, EM-PCA, multidimensional scaling, and locally linear embedding, to identity emotion of facial animations. But the application was not for realistic human faces.

In our method, we decided to complement some of these limitations of PCA by adopting the MPCA algorithm together with the LDA algorithm as the basis for the study [3, 15]. The MPCA algorithm disregards the traditional method which is based on two-dimensional data and uses instead vectors and integrates multiple face images into a high-dimensional tensor and processes data in tensor space. The advantage of this approach lies in its ability to persistently structure facial information images and consequently increases the accuracy rate when spatial relationships between pixels are considered. When the light brightness changes or facial expression changes, spatial structural information between pixels becomes particularly important.

LDA was adopted to further reduce the dimensions of samples processed by MPCA as it is capable of aggregating the samples in subspace and hence improving the face recognition rate [16, 17]. We combine MPCA and LDA to form LDA subspace, from which both MPCA features and LDA features can be extracted.

The organization of this paper is as follows. Our proposed algorithm will be discussed in Section 2. Methodology of the approach is presented in Section 3. To demonstrate the effectiveness of the proposed method, experimental results will be shown in Section 4. Finally conclusions are drawn in Section 5.

2. Principle of MPCA

In computer vision, most of the objects are naturally considered as nth-order tensors ($n \geq 2$) [18]. Take Figure 1 as an example; the image matrix in (i) is a 2nd-order tensor and a movie clip, while in (ii) it is a 3rd-order tensor. Traditional techniques for subspace dimensionality reduction such as PCA could transform image matrix to vectors with high dimensionality in one mode only, which cannot meet the need of dimensionality reduction. So such techniques are unable to handle multidimensional objects well and get satisfactory results. Therefore, in order to reduce dimensionality, a reduction algorithm which can directly operate on a high-order tensor object is desirable. Two-dimensional PCA (2DPCA) algorithm is proposed and developed, while researches are using dimensionality reduction solutions which represent facial images as matrices (2nd-order tensors) instead of vectors [19–22]. However, 2DPCA can only project images in single mode, which results in bad dimensionality of reduction [3, 23]. Thus, a more efficient algorithm MPCA has been proposed to get better dimensionality reduction.

2.1. Tensor Notations and Definitions. Multilinear principal component analysis (MPCA) has been introduced in details in [3], which is used to solve the problem of gait recognition. Before describing MPCA, the notations will be shown in this paper.

Vector α denotes 1st-order tensor. Matrix A denotes 2nd-order tensor. A_{ijk} denotes 3rd-order tensor. Higher-order tensors are indicated by $A_{i_1 i_2 \ldots i_n}$. Assume image matrix is indicated by $X \in R^{n_1 \times n_2}$. Tensor space is indicated by $R^{n_1} \times R^{n_2}$. $(u_1, u_2, \ldots, u_n)$ indicates the orthonormal bases of vector space R^{n_1} and $(v_1, v_2, \ldots, v_{n_2})$ indicates the orthonormal bases of vector space R^{n_2}. Vector $u_i v_j^T$ indicates orthonormal bases of tensor space $R^{n_1} \otimes R^{n_2}$. Image matrix X equals

$$X = \sum_{ij} \left(u_i^T X v_j\right) u_i v_j^T. \tag{1}$$

Define two matrices $U = [u_1, u_2, \ldots, u_{m_1}] \in R^{n_1 \times m_1}$ and $V = [v_1, v_2, \ldots, v_{m_1}] \in R^{n_2 \times m_2}$. Assume u, v indicate subspace of space R^{n_1}, R^{n_2} formed by basis vectors $\{u_i\}_i^{m_1} = 1$ and $\{v_j\}_j^{m_2} = 1$. Then $u \otimes v$ indicates subspace of tensor space $R^{n_1} \otimes R^{n_2}$. The result of 2nd-order tensor $X \in R^{n_1 \times n_2}$ projected to $u \otimes v$ is indicated by

$$Y = U^T X V \in R^{m_1 \times m_2}. \tag{2}$$

Based on different objective functions, transformation matrices U and V can be obtained by iteration; hence dimension reduction can be achieved.

2.2. Principle of MPCA. MPCA is developed based on the PCA algorithm. Its advantage is that it operates on tensor, replacing the traditional algorithms which transform high-dimensional data into one-dimensional vector. For example, to process 100 face images with size 112 × 92, PCA treats them as a 100 × 10304 matrix while MPCA treat them as a 100 × 112 × 92 tensor. MPCA have the advantage of taking into account correlation in the original data which is ignored by PCA.

Assume there are tensor sets of images $\{X_1, X_2, \ldots, X_M\}$; a tensor object is denoted by $X_m \in R^{I_1 \times I_2 \times \cdots \times I_N}$; I_n denotes dimensionality of n-order tensor. Each tensor can be unfolded as

$$X = S \times {}_1U^{(1)} \times {}_2U^{(2)} \times \cdots \times {}_NU^{(N)}. \tag{3}$$

Here, $U^{(n)}$ denotes orthogonal matrix. So, $\|X\|_F^2 = \|S\|_F^2$ [24]. Decompose this matrix; we can get

$$X_{(n)} = U^{(n)} \cdot S_{(n)} \cdot \left(U^{(n+1)} \otimes U^{(n+2)} \otimes \cdots \otimes U^{(N)} \otimes U^{(1)} \otimes U^{(2)} \otimes \cdots \otimes U^{(n-1)}\right)^T. \tag{4}$$

The key point of MPCA algorithm is to find a tensor subspace which can catch the variety of tensor objects and extract features of object. According to (4), projection of tensor samples onto tensor subspace is defined as

$$Y = X \times {}_1\widetilde{U}^{(1)^T} \times {}_2\widetilde{U}^{(2)^T} \times \cdots \times {}_N\widetilde{U}^{(N)^T}, \tag{5}$$

where Y denotes tensor after projection. $Y = \{Y_1, Y_2, \ldots, Y_M\}$, $Y_m \in R^{P_1 \times P_2 \times \cdots \times P_N}$. Figure 2 depicts the process.

As Figure 2 shows, by projecting each mode of facial tensor X, low-dimensional facial tensor which satisfies maximum variance can be achieved.

(i) Second-order tensor (ii) Third-order tensor

Figure 1: 2nd-order and 3rd-order tensor representations samples.

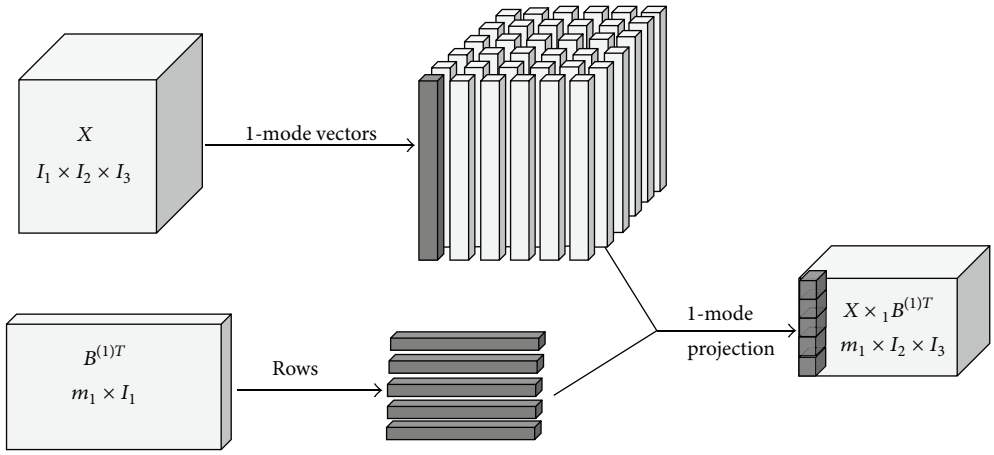


Figure 2: Illustration of the multilinear projection in the 1-mode vector space.

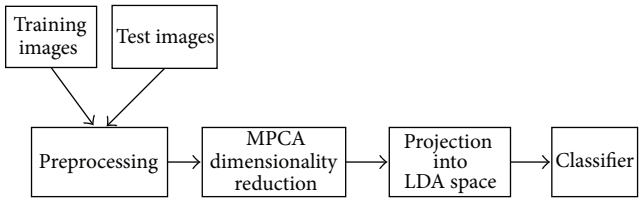


Figure 3: Flow chart of face recognition algorithm.

Figure 4: Face image examples of two persons in ORL face database.

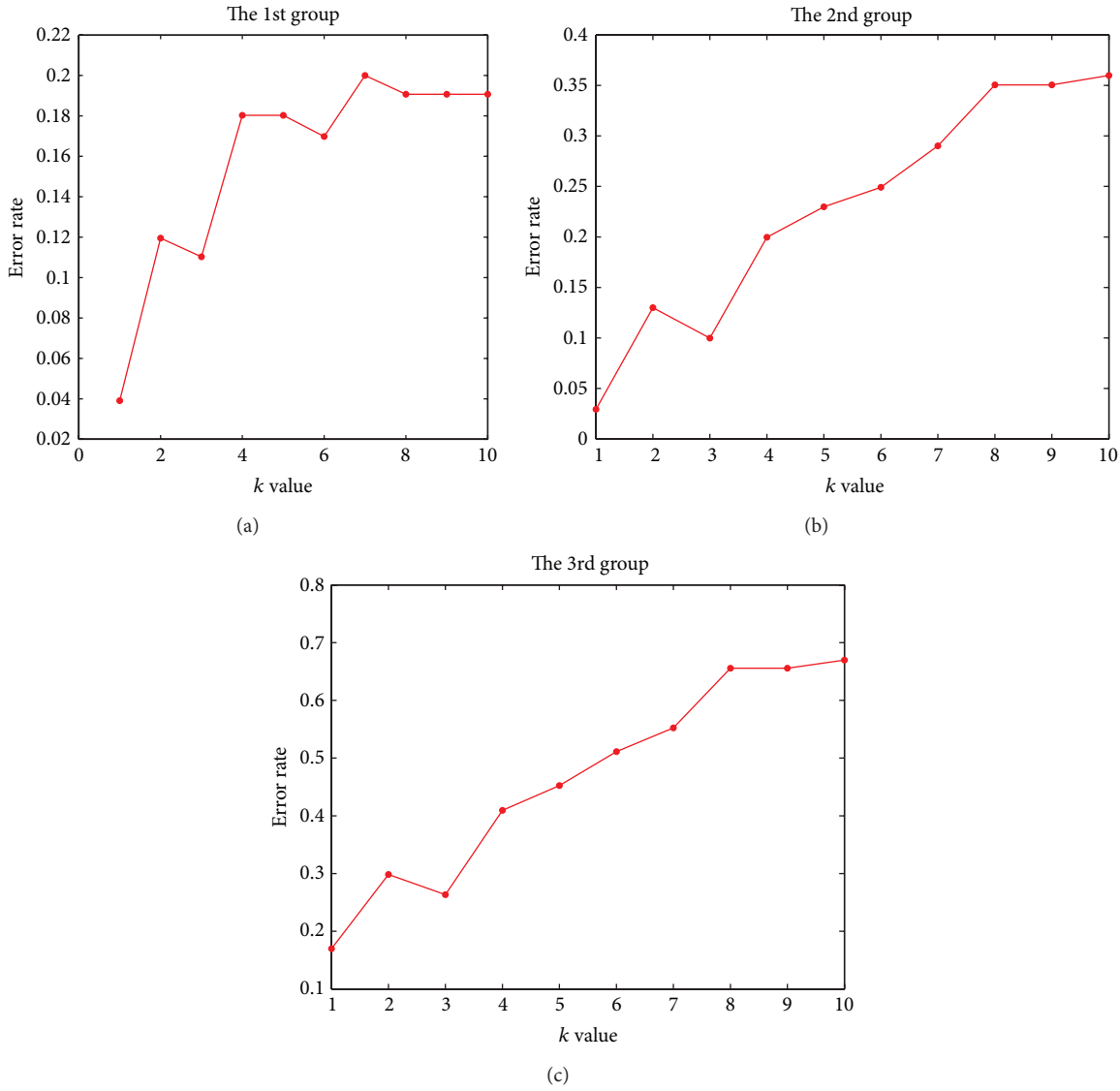


FIGURE 5: Recognition error rate of PCA against different k values (best LDA dimension reduction). In the 1st group, k equals 18. In the 2nd group, k equals 98. In the 3rd group, k equals 61.

For tensor objects of image samples, the variance before projection is as follows:

$$\Psi_X = \sum_{m=1}^{M} \left\| X_m - \overline{X} \right\|_F^2, \qquad \overline{X} = \left(\frac{1}{M} \right) \sum_{m=1}^{M} X_m. \tag{6}$$

And the tensors after projection satisfy the following equation:

$$\Psi_Y = \sum_{m=1}^{M} \left\| Y_m - \overline{Y} \right\|_F^2, \qquad \overline{Y} = \left(\frac{1}{M} \right) \sum_{m=1}^{M} Y_m. \tag{7}$$

By combining (5) and (6), we can get the following equation:

$$\psi_Y = \sum_{m=1}^{M} \left\| X_m \times \widetilde{U}^{(1)^T} \times \widetilde{U}^{(2)^T} \times \cdots \times \widetilde{U}^{(N)^T} - \overline{x} \times \widetilde{U}^{(1)^T} \times \widetilde{U}^{(2)^T} \times \cdots \times \widetilde{U}^{(N)^T} \right\|_F^2. \tag{8}$$

The MPCA algorithm equals to the resolving optimization problem:

$$\left\{ \widetilde{U}^{(1)},\ n = 1, 2, \ldots, N \right\} = \arg\max \psi_Y. \tag{9}$$

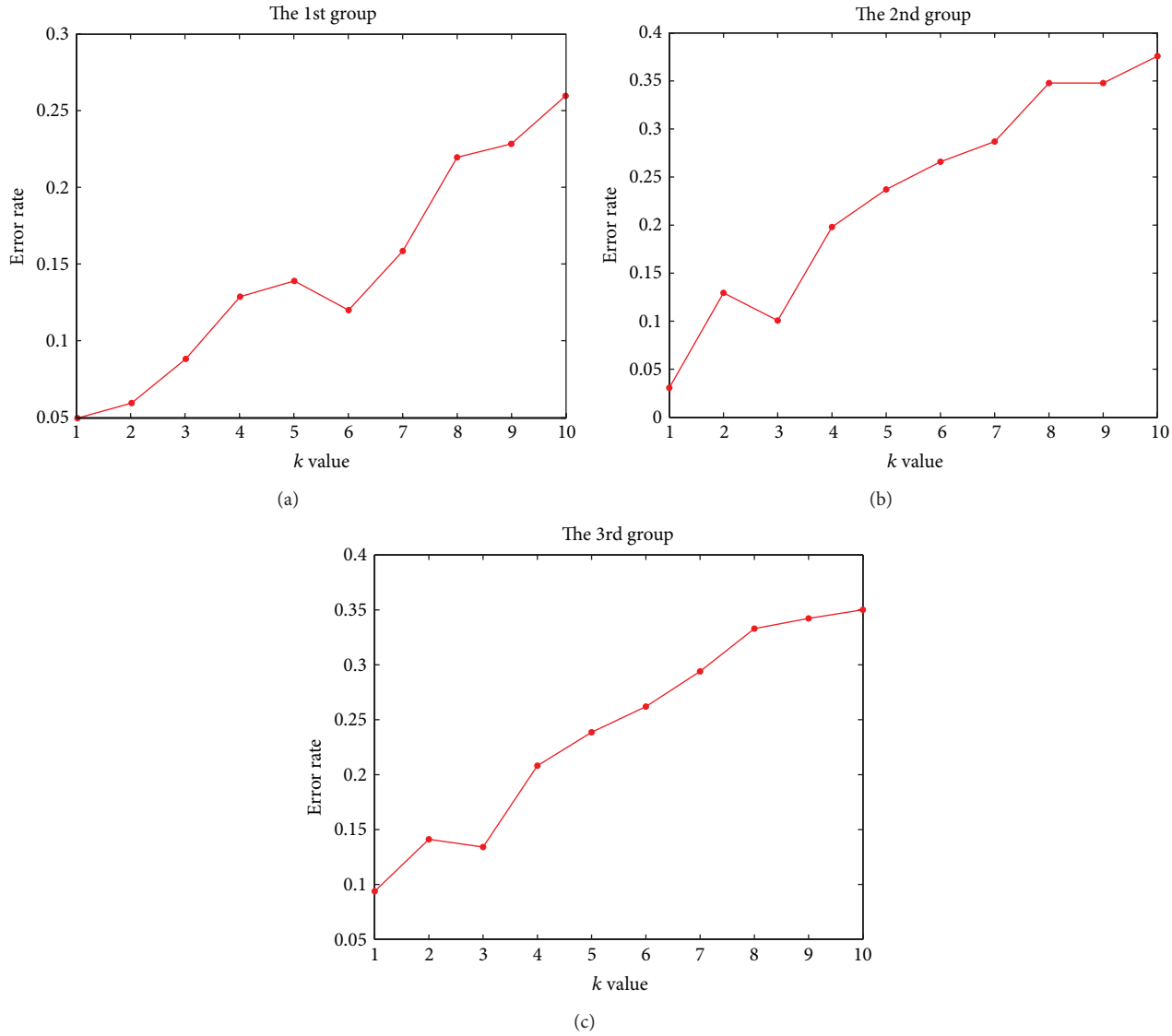


Figure 6: Recognition error rate of MPCA against different k values.

In (9), by using alternating-least-square method (ALS), we are able to calculate local optimization procedure. When solving the nth projection matrix $\widetilde{U}^{(n)}$, other matrices were set constant; tensor X is projected to tensor space $(R_1, \ldots, R_{n-1}, R_{n+1}, \ldots, R_N)$, where $\widetilde{U}_{j+1}^{(n)} = X \times \widetilde{U}_{j+1}^{(1)^T} \times \cdots \times \widetilde{U}_{j+1}^{(n-1)^T} \times \widetilde{U}_{j+1}^{(n+1)^T} \times \widetilde{U}_{j+1}^{(N)^T}$.

Column of $\widetilde{U}^{(n)}$ can be obtained from orthogonal basis of projection subspace. Sample X_m in (8) is projected to lower dimensional tensor $Y_m \in R^{P_1 \times P_2 \times \cdots \times P_{n-1} \times I_n \times P_{n+1} \cdots \times P_N}$. $Y_m^{(n)}$, nth-mode unfolding matrix of Y_m, is inputted to get PCA. It equals to

$$\arg\max \sum_{m=1}^{M} \left\| \widetilde{U}^{(n)^T} Y_m^{(n)} - \widetilde{U}^{(n)^T} \overline{Y}^{(n)} \right\|. \tag{10}$$

2.3. MPCA Algorithm. MPCA have managed to handle multidimensional objects. According to the above sections, pseudocode for the computation of the MPCA algorithm can be concluded [25] as shown in Figure 3.

Step 1. Input sample images and center them as $\{x_n \in R^{P_1 \times P_2}, n = 1, \ldots, N\}$.

Step 2. Obtain the total scatter matrix's eigendecomposition.

Step 3. Calculate the eigenvectors and their corresponding most significant eigenvalues, and the result is output as $\widetilde{U}^{(n)}$.

Step 4. (i) Get $\{\widehat{y}_n = \widetilde{U}^{(1)^T} \times \widetilde{x}_n \times \widetilde{U}^{(2)}, n = 1, \ldots, N\}$.

(ii) Calculate $\psi_{Y_0} = \sum_{n=1}^{M} \|\widehat{y}_n\|_F^2$.

(iii) For $k = 1 : K$,

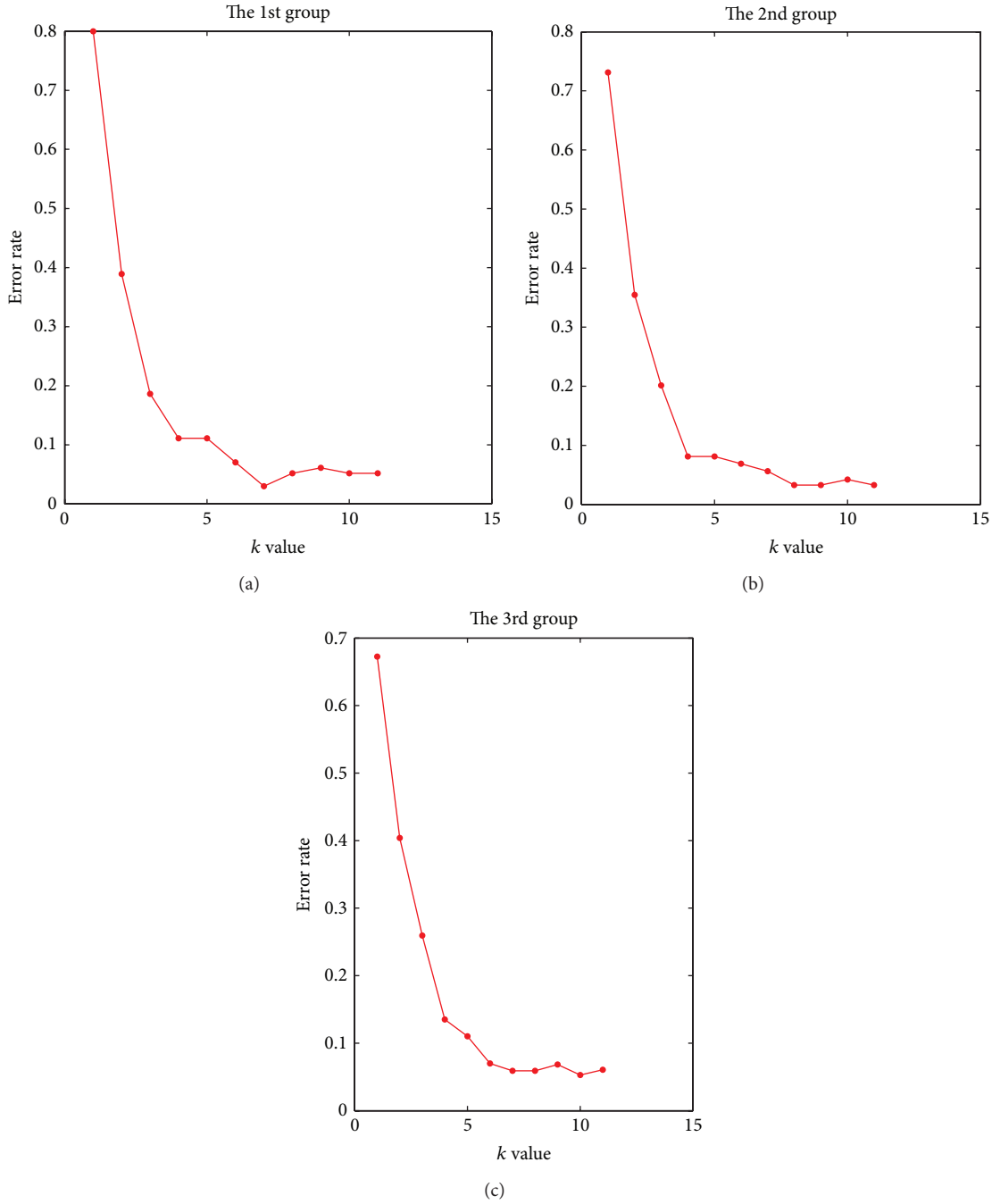


Figure 7: Recognition error rate of PCA + LDA against different LDA dimension reduction values.

(a) calculate the total scatter matrix's eigenvectors and their corresponding most significant eigenvalues, and the result is output as $\widetilde{U}^{(j)}$, for $j = 1, 2$;

(b) get ψ_{Y_k} and $\{\hat{y}_n, n = 1, \ldots, N\}$;

(c) if $\psi_{Y_k} - \psi_{Y_{k-1}} < j$, then break the loop and go to Step 5.

Step 5. Finally calculate the feature matrix; see the following equation:

$$y_n = \widetilde{U}^{(1)^T} \times \tilde{x}_m \times \widetilde{U}^{(2)}, \quad n = 1, \ldots, N. \tag{11}$$

2.4. LDA Algorithm. LDA (linear discriminant analysis) projects image onto a lower-dimensional vector space to achieve maximum discrimination as follows.

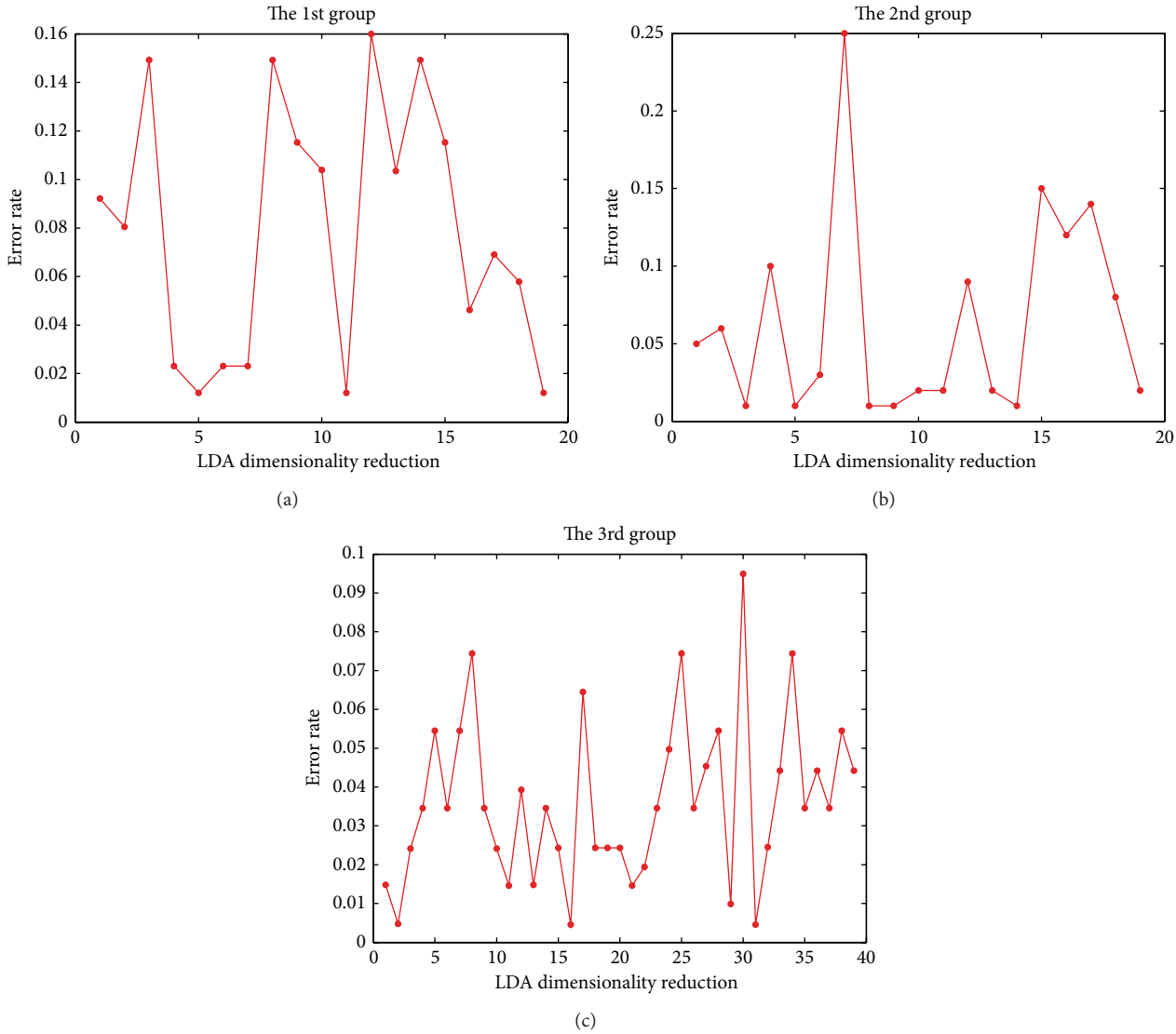


Figure 8: Error rate of MPCA + LDA algorithm against different LDA dimension reduction values.

Step 1. Compute the average sample values for different kinds of facial images in the original space. Total number is denoted by c. X_{ij} denotes the jth object of the ith class of samples:

$$\overline{m_i} = \frac{1}{n_i}\sum_{j=1}^{n_1} X_{ij}, \qquad X_{ij} \in R^d, \quad i = 1, 2, \ldots, c,$$
$$m = \sum_{i=1}^{c} p_i m_i. \tag{12}$$

Step 2. Compute covariance matrix of each class:

$$c_i = \frac{1}{n_i}\sum_{j=1}^{n_i} \left(X_{ij} - m_i\right) \cdot \left(X_{ij} - m_i\right)^T. \tag{13}$$

Step 3. Compute within-class and between-class scatter matrices:

$$C_b = \sum_{i=1}^{c} p_i \left(m_i - m\right) \cdot \left(m_i - m\right)^T,$$
$$C_w = \sum_{i=1}^{c} c_i. \tag{14}$$

Step 4. Compute eigenvectors of matrix $C_w^{-1}C_b$ to get projection vectors. Then dimensionality reduction data can be obtained by projection [26, 27].

After dimensionality reduction using MPCA, the matrices are arranged in columns into vectors as inputs to the LDA algorithm. By using MPCA algorithm to reduce the dimension of the image, we not only solved the problem of

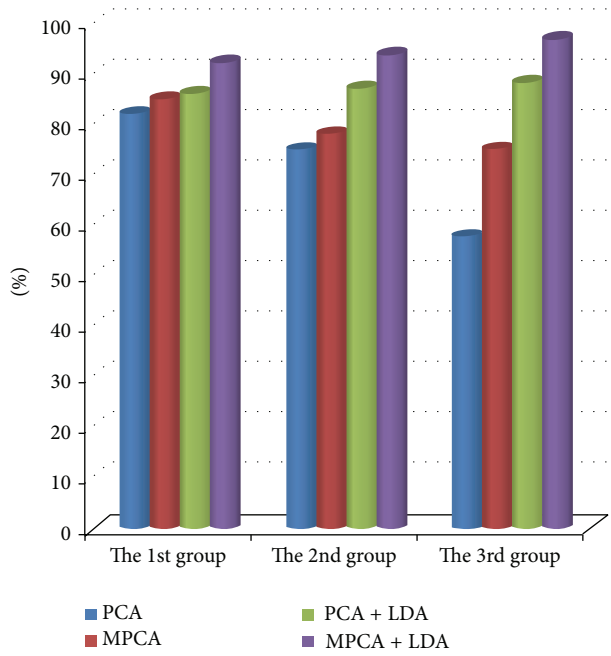


Figure 9: Histogram of recognition results in experiments.

singular matrix but also retained structure information in the images and thus improve the recognition rate.

2.5. KNN Algorithm. K-nearest neighbor (KNN) algorithm [28, 29] is adopted for sample set classification here, and the concrete steps are as follows.

Step 1. Select different parameters of K value.
Step 2. Adopt the method of cross-validation on training face images, for $k = 1 : K$.

Step 3. Make the cross-validation error classification rate minimization and get its corresponding parameter k.

Step 4. Construct a prediction model with k.

3. Process of the Recognition Algorithm

3.1. Preprocessing. Image preprocessing and normalization are vital for face recognition systems as images are often affected by image quality, illumination, face rotation, facial expression [8, 30], and so forth. In order to offset above factors, it is necessary for us to carry out face normalization before facial feature extraction.

Our data is preprocessing normalized images with a resolution of 80×80. In our research, histogram equalization was applied (see (15)):

$$y' = \frac{y - \min(y)}{\max(y) - \min(y)}. \tag{15}$$

3.2. Dimensionality Reduction Using MPCA and Feature Matrix Extraction Using LDA. MPCA reduces dimensions of input face images and generates feature projection matrix [30] that are then taken as input samples to LDA. MPCA and LDA combination were used to construct LDA subspace, from which both MPCA features and LDA features can be extracted.

The detailed steps have been described in Sections 2.2 and 2.3.

3.3. Face Recognition Using L2 Distance Measure. We used resultant output acquired above as input samples for training and applied aforesaid techniques to get the feature matrix. Then we carried out a similarity measure on image samples. In our research, we choose L2 distance for measures (see (16)):

$$d(a, b) = \sqrt{\sum_{h=1}^{H} [a(h) - b(h)]^2}. \tag{16}$$

KNN classifier [31] is adopted for sample set classification here, while the procedure and details are introduced in Section 2.4.

The overall approach of face recognition proposed in this paper is shown in Figure 3.

4. Experiments

We evaluated the performance of our algorithm based on MPCA + LDA in this research and compared with the PCA, MPCA, and PCA + LDA algorithm by performing experiments on ORL databases [32]. In order to examine the ability of our method, we also try it on other classical face databases such as FERET and YALE.

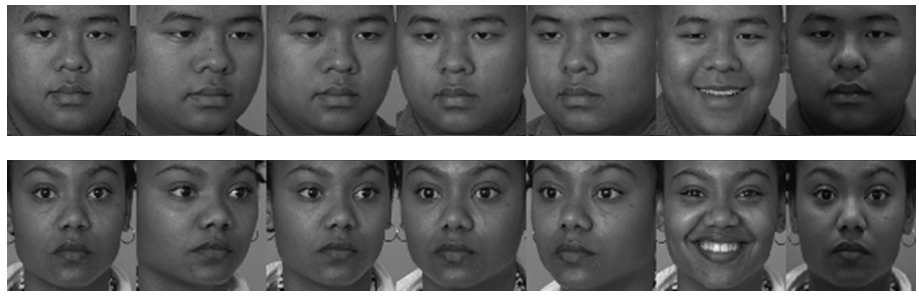


Figure 10: Face image examples of two persons in FERET face database.

Figure 11: Face image examples of two persons in YALE face database.

The experiments were conducted with three groups. We choose part of images in each group for training, while the rest for testing. As the probabilities for each kind of facial samples are the same, then P_i that equals 1 is set in LDA algorithm.

Initially, we tested how different parameters affect the recognition error rate and how classification result is affected by dimensionality using the MPCA, dimensionality after using the LDA and k value of KNN algorithm. LDA algorithm requires dimension reduction not greater than the total number of samples minus 1, so $1 \leq$ LDA dimension reduction ≤ 19. There are 10 samples in each category, so $1 \leq k \leq 10$. Other parameters of MPCA are set to the optimized values.

4.1. Experiments on the ORL. The ORL face database contains a total of 400 images of 40 individuals (each individual has 10 gray scale images) [33]. Some photos are taken in different periods, and some are taken with the various countenances and the facial details. Each image is of a resolution of 256 grey levels per pixel [34]. Figure 4 shows image examples of two persons before preprocessing.

Now, images have been divided into three different groups. With the first group, we select the first 5 images of the first 20 persons as training data and the last 5 images of the first 20 persons as test samples for face identification. With the second group we select the first 5 images of the rest 20 persons as training data and the last 5 images of the rest 20 persons as testing samples. With the third group we select the first 5 images of 40 persons as training data and the last 5 images of 40 persons as testing samples.

Recognition error rate of PCA is shown in Figure 5.

Judging from the figure, when k equals 1, the recognition error rate reaches minimal value. PCA recognition accuracy reaches 58%–82%.

Error rate of MPCA under different k value is shown in Figure 6. As shown in the figure, when k equals 1, error rate is minimal. MPCA recognition accuracy reaches 75%–85% in the experiments.

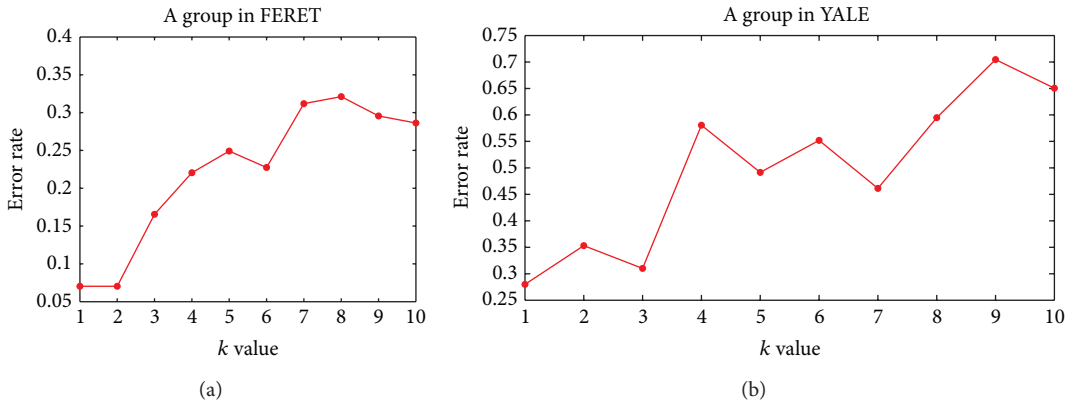
When k equals 8, error rate of PCA + LDA algorithm reaches minimal value 7%. How different LDA dimension reduction affects recognition accuracy is shown in Figure 7.

We can see from Figures 5, 6, and 7 that dimension after LDA increases as the number of samples also increases. When applying PCA + LDA algorithm, we use MPCA to decrease dimension of facial samples to 11 and then use LDA. For the 1st group, reduce dimension to 7. For the 2nd group, reduce dimension to 8. For the 3rd group, reduce dimension to 10. We can conclude that the LDA algorithm is not satisfied with multidimensional objects. Accuracy of PCA + LDA reaches 86%–88%.

MPCA + LDA algorithm only produces higher error rate of 10%–25% when k equals 10. In other situations, the recognition error rate is very low. When k equals 8, the error rate of different LDA dimensionality reduction is shown in Figure 8. Algorithm recognition accuracy rate reaches a high value.

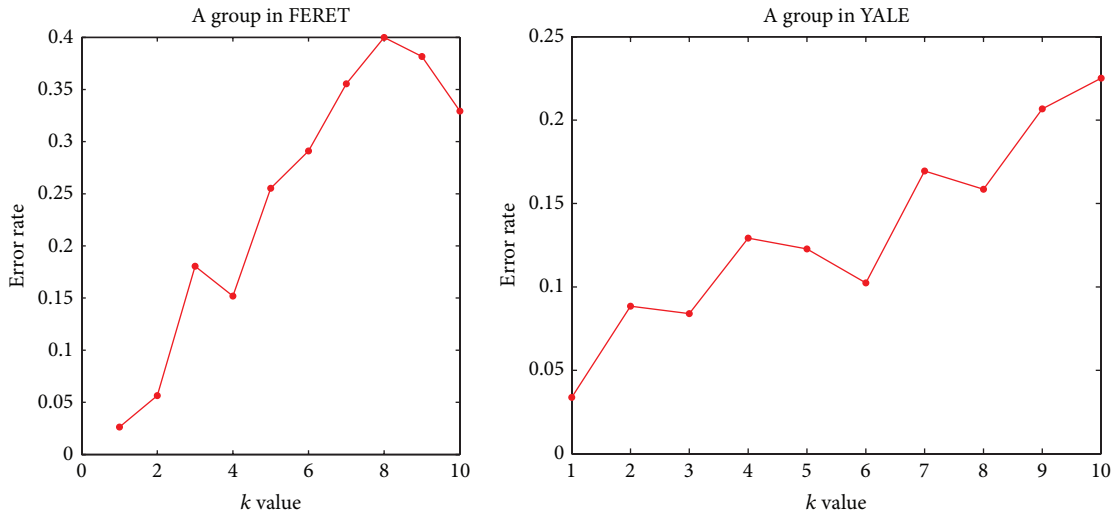
Result of the experiments on ORL database is shown in Figure 9. Take recognition accuracy of four algorithms for comparison; the combination of MPCA and LDA does result in better recognition performance than traditional methods.

4.2. Experiments on More Face Databases. We choose FERET and YALE for our experiments. Implement steps are similar



(a) (b)

Figure 12: Recognition error rate of PCA against different k values (best LDA dimension reduction). In one group of FERET, k equals 27. In one group of YALE, k equals 63.



(a) (b)

Figure 13: Recognition error rate of MPCA against different k values.

to those in above section, so we just simplify steps and focus on the results.

FERET face database consists of a total of 1400 images of 200 individuals (each person has 7 different images). Figure 10 shows image examples of two persons before preprocessing.

YALE face database contains 165 images of 15 individuals; Figure 11 shows image examples of two persons before preprocessing [35].

The performance of PCA, MPCA, PCA plus LDA, and MPCA + LDA techniques is tested by varying the number of eigenvectors. We have chosen one group of result in each database for comparison.

PCA performed worse on YALE than on FERET because of the poor adaptability for the image brightness and facial expression, which is shown in Figure 12.

Though in Figure 13 MPCA performed much well on face recognition in YALE database, the process takes longer time than with PCA.

Figures 14 and 15 show that both PCA + LDA and MPCA + LDA can turn to high accuracy and low error rate in recognition. However, PCA + LDA effectively sees only the Euclidean structure, while MPCA + LDA successes to discover the underlying structure [36].

Compared against all the other algorithms, although with simple preprocessing, we can learn that MPCA + LDA has achieved best overall performance in both FERET and YALE databases.

5. Conclusions

This paper presents an algorithm for face recognition based on MPCA and LDA. As opposed to other traditional methods, our proposed algorithm treats data as multidimensional tensor and fully considers the spatial relationship. The advantage of our approach is of great relevance to applications and is capable of recognizing face dataset under different lighting

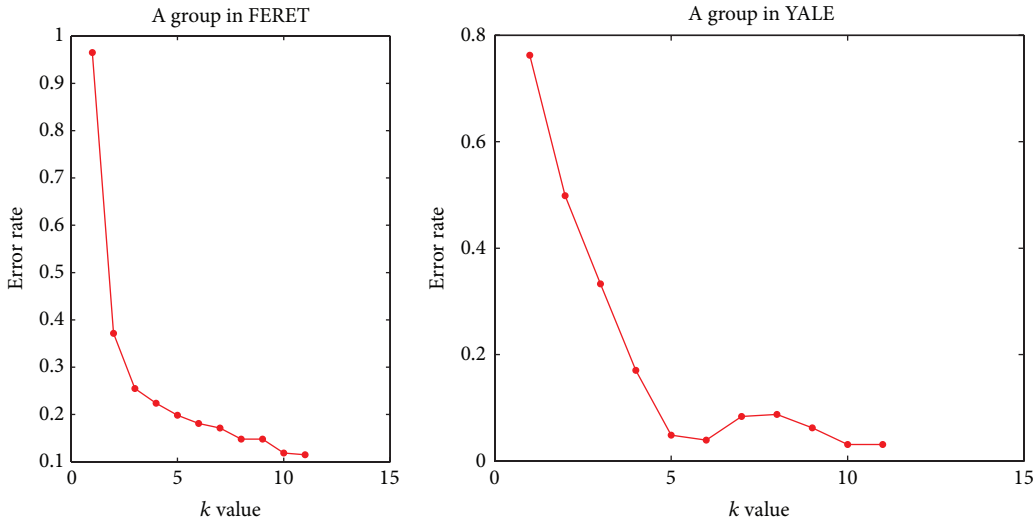


Figure 14: Recognition error rate of PCA + LDA against different LDA dimension reduction values.

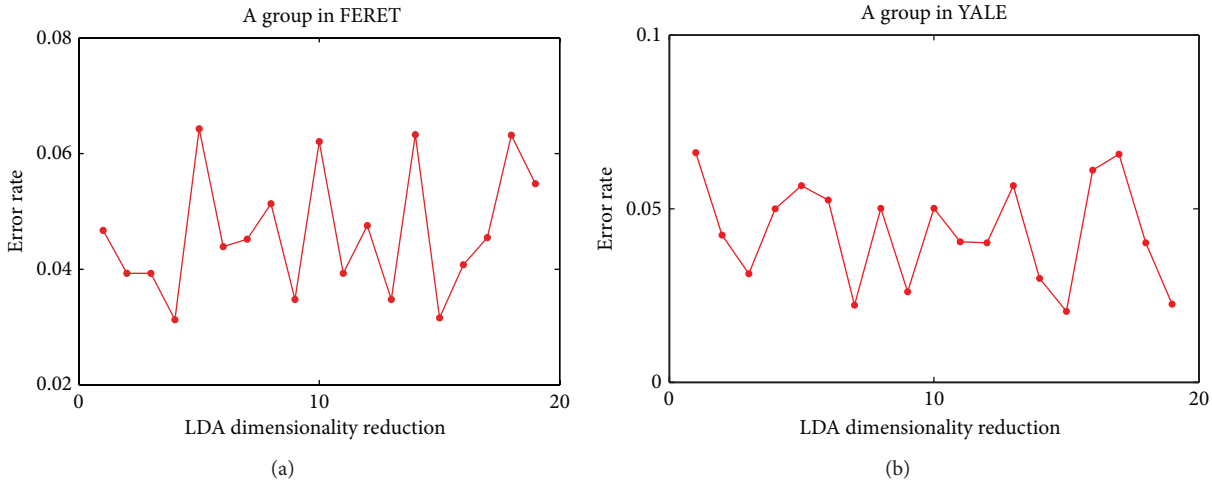


Figure 15: Error rate of MPCA + LDA algorithm against different LDA dimension reduction values.

conditions and with various facial expressions. LDA algorithm projects the data to a new space and has exact clustering result in our experiments. Compared with traditional face recognition algorithms, our proposed algorithm is not only a boost in recognition accuracy but also an unclogging of dimensionality bottlenecks and an efficient resolution of the small sample size problem. Future work of our research will include applying this approach on larger face databases such as on the CMU Multi-PIE, NIST's FRGC, and MBGC.

Conflict of Interests

The authors declared that there is no conflict of interests regarding the publication of this paper.

Acknowledgments

The authors would like to show an appreciation of reviewers' insightful and constructive comments and would like to thank everyone for their hard work on this research. The work was supported by a Grant from the Ph.D. Programs Foundation of Ministry of Education of China (no. 20120141120006), Hubei Planning Project of Research and Development (no. 2011BAB035), and Wuhan Planning Project of Science and Technology (no. 2013010501010146).

References

[1] F. Song, H. Liu, D. Zhang, and J. Yang, "A highly scalable incremental facial feature extraction method," *Neurocomputing*, vol. 71, no. 10-12, pp. 1883–1888, 2008.

[2] W. Zhao, R. Chellappa, P. J. Phillips, and A. Rosenfeld, "Face recognition: a literature survey," *ACM Computing Surveys*, vol. 35, no. 4, pp. 399–458, 2003.

[3] H. Lu, K. N. Plataniotis, and A. N. Venetsanopoulos, "MPCA: multilinear principal component analysis of tensor objects," *IEEE Transactions on Neural Networks*, vol. 19, no. 1, pp. 18–39, 2008.

[4] S. Fernandes and J. Bala, "Performance analysis of PCA-based and LDA-based algorithms for face recognition," *International Journal of Signal Processing Systems*, vol. 1, no. 1, pp. 1–6, 2013.

[5] B. A. Draper, K. Baek, M. S. Bartlett, and J. R. Beveridge, "Recognizing faces with PCA and ICA," *Computer Vision and Image Understanding*, vol. 91, no. 1-2, pp. 115–137, 2003.

[6] K. Choudhary and N. Goel, "A review on face recognition techniques," in *Proceedings of the International Conference on Communication and Electronics System Design*, International Society for Optics and Photonics, 2013.

[7] R. Gottumukkal and V. K. Asari, "An improved face recognition technique based on modular PCA approach," *Pattern Recognition Letters*, vol. 25, no. 4, pp. 429–436, 2004.

[8] J. Li, B. Zhao, and H. Zhang, "Face recognition based on PCA and LDA combination feature extraction," in *Proceedings of the 1st International Conference on Information Science and Engineering (ICISE '09)*, pp. 1240–1243, IEEE, December 2009.

[9] P. Viola and M. J. Jones, "Robust real-time face detection," *International Journal of Computer Vision*, vol. 57, no. 2, pp. 137–154, 2004.

[10] X. R. L. Y. Tang Liang, "An face recognition technique based on discriminative common vector in PCA transform space," *Journal of Wuhan University*, vol. 34, no. 4, 2009.

[11] H. Lu, K. N. Plataniotis, and A. N. Venetsanopoulos, "A survey of multilinear subspace learning for tensor data," *Pattern Recognition*, vol. 44, no. 7, pp. 1540–1551, 2011.

[12] A. K. Bansal and P. Chawla, "Performance evaluation of face recognition using PCA and N-PCA," *International Journal of Computer Applications*, vol. 76, no. 8, pp. 14–20, 2013.

[13] J. F. Pereira, R. M. Barreto, G. D. C. Cavalcanti, and T. I. Ren, "A robust feature extraction algorithm based on class-modular image principal component analysis for face verification," in *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP '11)*, 2011.

[14] F. S. Tsai, "Dimensionality reduction for computer facial animation," *Expert Systems with Applications*, vol. 39, no. 5, pp. 4965–4971, 2012.

[15] H. Lu, K. N. Plataniotis, and A. N. Venetsanopoulos, "Uncorrelated multilinear discriminant analysis with regularization and aggregation for tensor object recognition," *IEEE Transactions on Neural Networks*, vol. 20, no. 1, pp. 103–123, 2009.

[16] W. Zhao, R. Chellappa, and N. Nandhakumar, "Empirical performance analysis of linear discriminant classifier," in *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pp. 164–169, IEEE, June 1998.

[17] Y. Xie, "LDA algorithm and its application to face recognition," *Computer Engineering and Applications*, vol. 46, no. 19, pp. 189–192, 2010.

[18] S. Yan, D. Xu, Q. Yang, L. Zhang, and H. Zhang, "Multilinear discriminant analysis for face recognition," *IEEE Transactions on Image Processing*, vol. 16, no. 1, pp. 212–220, 2007.

[19] D. Zhang and Z. Zhou, "(2D)2 PCA: two-directional two-dimensional PCA for efficient face representation and recognition," *Neurocomputing*, vol. 69, no. 1–3, pp. 224–231, 2005.

[20] D. Zhang, X. You, P. Wang, S. N. Yanushkevich, and Y. Y. Tang, "Facial biometrics using nontensor product wavelet and 2d discriminant techniques," *International Journal of Pattern Recognition and Artificial Intelligence*, vol. 23, no. 3, pp. 521–543, 2009.

[21] J. Yang, D. Zhang, and A. F. Frangi, "Two-dimensional PCA: a new approach to appearance-based face representation and recognition," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 26, no. 1, pp. 131–137, 2004.

[22] Y. Li, H. Xie, and Y. Zhou, "Study of eyebrow recognition based on 2 DPCA," *Journal of Wuhan University*, vol. 57, no. 6, pp. 517–522, 2011.

[23] H. Lu, K. N. Plataniotis, and A. N. Venetsanopoulos, "Uncorrelated multilinear principal component analysis through successive variance maximization," in *Proceedings of the 25th International Conference on Machine Learning*, pp. 616–623, July 2008.

[24] L. De Lathauwer, B. De Moor, and J. Vandewalle, "On the best rank-1 and rank-$(R_1 R_2, ..., R_n)$ approximation of higher-order tensors," *SIAM Journal on Matrix Analysis and Applications*, vol. 21, no. 4, pp. 1324–1342, 2000.

[25] C. Chen, S. Zhang, and Y. Chen, "Face recognition based on MPCA," in *Proceedings of the 2nd International Conference on Industrial Mechatronics and Automation (ICIMA '10)*, pp. 322–325, Wuhan, China, May 2010.

[26] P. N. Belhumeur, J. P. Hespanha, and D. J. Kriegman, "Eigenfaces versus fisherfaces: recognition using class specific linear projection," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 19, no. 7, pp. 711–720, 1997.

[27] X. Su, Q. Zeng, and X. Wang, "Several combination methods of face recognition based on PCA and LDA," *Computer Engineering and Design*, vol. 33, no. 9, pp. 3574–3578, 2012.

[28] G.-F. Lu, Y. J. Wang, and J. Zou, "Improved complete neighbourhood preserving embedding for face recognition," *IET Computer Vision*, vol. 7, no. 1, pp. 71–79, 2013.

[29] E. Nasibov and C. Kandemir-Cavas, "Efficiency analysis of KNN and minimum distance-based classifiers in enzyme family prediction," *Computational Biology and Chemistry*, vol. 33, no. 6, pp. 461–464, 2009.

[30] J. Shermina, "Face recognition system using multilinear principal component analysis and locality preserving projection," in *Proceedings of the IEEE GCC Conference and Exhibition (GCC '11)*, pp. 283–286, IEEE, February 2011.

[31] Y. Liaw, M. Leou, and C. Wu, "Fast exact k nearest neighbors search using an orthogonal search tree," *Pattern Recognition*, vol. 43, no. 6, pp. 2351–2358, 2010.

[32] F. S. Samaria and A. C. Harter, "Parameterisation of a stochastic model for human face identification," in *Proceedings of the 2nd IEEE Workshop on Applications of Computer Vision*, pp. 138–142, Sarasota, Fla, USA, December 1994.

[33] Y. Jin and Q. Ruan, "Orthogonal locality sensitive discriminant analysis for face recognition," *Journal of Information Science and Engineering*, vol. 25, no. 2, pp. 419–433, 2009.

[34] P. P. Paul and M. Gavrilova, "Multimodal cancelable biometrics," in *Proceedings of the IEEE 11th International Conference on Cognitive Informatics & Cognitive Computing (ICCI '12)*, 2012.

[35] P. Punitha and D. S. Guru, "Symbolic image indexing and retrieval by spatial similarity: an approach based on B-tree," *Pattern Recognition*, vol. 41, no. 6, pp. 2068–2085, 2008.

[36] X. He, S. Yan, Y. Hu, P. Niyogi, and H. Zhang, "Face recognition using Laplacianfaces," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 27, no. 3, pp. 328–340, 2005.

Hindawi Publishing Corporation
Mathematical Problems in Engineering
Volume 2014, Article ID 972304, 11 pages
http://dx.doi.org/10.1155/2014/972304

Research Article

3D Maps Representation Using GNG

Vicente Morell, Miguel Cazorla, Sergio Orts-Escolano, and Jose Garcia-Rodriguez

University Institute for Computing Research, University of Alicante, P.O. Box 99, 03080 Alicante, Spain

Correspondence should be addressed to Miguel Cazorla; miguel.cazorla@ua.es

Received 6 March 2014; Accepted 24 July 2014; Published 27 August 2014

Academic Editor: Yi Chen

Current RGB-D sensors provide a big amount of valuable information for mobile robotics tasks like 3D map reconstruction, but the storage and processing of the incremental data provided by the different sensors through time quickly become unmanageable. In this work, we focus on 3D maps representation and propose the use of the Growing Neural Gas (GNG) network as a model to represent 3D input data. GNG method is able to represent the input data with a desired amount of neurons or resolution while preserving the topology of the input space. Experiments show how GNG method yields a better input space adaptation than other state-of-the-art 3D map representation methods.

1. Introduction

A 3D point is comprised of (X, Y, and Z) values representing the spatial coordinates. Moreover, if color information (R, G, and B) is available for each point, it is referred to as RGB-D data. RGB-D cameras provide this kind of data and have become very popular due to their low cost, like the Kinect sensor. This sensor could provide more than 300,000 3D points per capture. That feature is very useful to carry out a fundamental task in mobile robotics: mapping [1]. Mapping is a task that builds a map from the observations and movements of the robot. Each time the robot moves, an observation is linked to that movement. Then, using different methods, for example, registration, the map can be built, transforming each observation with respect to a common coordinates frame. This map is useful to develop subsequent tasks, like localization, navigation, and recognition. The use of RGB-D data as observations is referred to as RGB-D mapping and RGB-D maps.

The amount of data in RGB-D maps is huge since the number of poses is high. In a typical map with 10,000 poses, the data could consist of more than 3 billion of 3D points, which are unaffordable for representation and other tasks. Furthermore, as acquisitions frame rate is high, for a common area, a huge amount of redundant points is used to represent the input space. Due to the huge quantity of data, several methods have been proposed to reduce the number of points in the map while preserving the main features of the data, as it would be used in further tasks.

Elevation maps were a commonly used structure in the past [2]. These elevation maps are represented using a regular 2D cell grid where each cell value represents the elevation or height of the surface of that space. This compact model allows a simple representation of large areas but with low level of detail. Triebel et al. [3] presented an extension of the height maps to represent different surfaces at different heights. This multilevel surface map (MLS map) allows the representation of vertical structures and different surfaces in a 2D cell-based structure like the ones used in the traditional height maps. This approach focuses on the representation of planar surfaces to help mobile robotics tasks like robotics navigation.

Following this idea of 3D space representation, some other structures have been proposed like occupancy grids or Octrees. Occupancy grids represent the entire space as 3D cell grids. The cell information could consist of a single value

of occupancy or contain more complex information as the probability of occupancy. Several works in mobile robotics have used this structure as a base for their applications [4–6]. Another common structure is the Octree [7]. The Octree is a tree structure in which each internal node has eight children. Each node of the tree is recursively subdivided into eight new nodes until a certain condition is fulfilled, like the size of the area represented by a node. This structure represents both occupied and empty space in the area represented by the Octree. It also allows some optimized operations like closest point searching or occupancy checking. In [8], an Octree based framework called OctoMap is presented. It uses probabilistic occupancy estimation where areas of the space are represented as occupied, empty, or uncertain. Another commonly used structure is the voxel grid (VG). The VG downsampling technique is based on the input space sampling using a grid of 3D voxels. This technique has been traditionally employed in the area of computer graphics to subdivide the input space and reduce the number of points [7, 9].

Wang et al. [10] presented a feature based 3D point cloud simplification method. They detect the points with more information (big curvature) and subsample the rest of the points using a uniform spherical sampling method. Therefore, they preserve the keypoints and subsample those points with less curvature information. This method is able to subsample 3D point clouds obtained from object surfaces. It does not work on scene maps since the spherical sampling and the feature selection process is usually harder and problem dependent.

Other approaches use self-organizing maps in order to reduce the input space. Viejo et al. [11] used a Growing Neural Gas (GNG) algorithm to filter and reduce single frontal point clouds. In this paper, we propose the extension of that work to manage complete maps. The GNG adapts its structure to the complete map, reducing its size, preserving the input space topology, and providing better adjustment than existing methods. To validate our method, we present several experiments comparing our method with map size reduction state-of-the-art methods.

The rest of this work is organized as follows. First, in Section 2 we introduce and describe the proposed GNG application and the Octree and voxel grid methods that we will use in the experimentation. Next, in Section 3, the validation of our method is carried out comparing it with the two previous mentioned methods. Finally, conclusions and future works are drawn.

2. 3D Representation Methods

One way of selecting points of interest in 3D point clouds is the use of a topographic mapping where a low-dimensional map is fitted to the high dimensional manifold of the model, whilst preserving the topographic structure of the data.

In this section, we review some typical methods to represent and compress 3D data. First, we propose the use of a Growing Neural Gas algorithm to reduce and represent 3D point cloud maps. Then, we briefly describe two well-known data structures in order to compare them with our method.

2.1. GNG Method. A common way to achieve data dimension reduction is by using self-organising neural networks where input patterns are projected onto a network of neural units such that similar patterns are projected onto units adjacent in the network and vice versa. As a result of this projection, a representation of the input patterns is achieved, which in postprocessing stages allows exploiting the similarity relations of the input patterns.

However, most common approaches do not provide good neighborhood and topology preservation if the logical structure of the input pattern is not known a priori. In fact, the most common approaches specify in advance the number of neurons in the network and a graph that represents topological relationships between them, for example, a two-dimensional grid, and seek the best match to the given input pattern manifold. When this is not the case, the networks fail to provide good topology preservation as in the case of Kohonen's algorithm [12].

The approach presented in this paper is based on self-organising networks trained using the Growing Neural Gas learning method [13], an incremental training algorithm. The links between the neurons in the network are established through competitive Hebbian learning [14]. As a result, the algorithm is suitable in cases where the topological structure of the input pattern is not known a priori and yields topology preserving maps of feature manifold [15].

In GNG, the nodes of the network compete to determine the ones with the highest similarity to the input distribution. In our case, the input distribution is a finite set of 3D points extracted from different types of sensors. The highest similarity reflects which node together with its topological neighbors is the closest one to the input sample point which is the pattern generated by the network. The n-dimensional input signals are randomly generated from a finite input distribution.

The nodes move towards the input distribution by adapting their position to the input data geometry. During the learning process local error measures are gathered to determine where to insert new nodes. New nodes are inserted near the node with the highest accumulated error. At each adaptation step a connection between the winner and its topological neighbors is created as dictated by the competitive Hebbian learning method. This is continued until an ending condition is fulfilled, for example, the evaluation of the optimal network topology, a predefined network size or a deadline.

The network is specified as follows.

(i) It is a set N of nodes (neurons). Each neuron $c \in N$ has its associated reference vector $w_c \in R^d$. The reference vectors can be regarded as positions in the input space of their corresponding neurons.

(ii) It is a set of edges (connections) between pairs of neurons. These connections are not weighted and its purpose is to define the topological structure. The edges are determined using the competitive Hebbian learning algorithm. An edge-aging scheme is used to remove connections that are invalid due to the activation of the neuron during the adaptation process.

The GNG learning algorithm is as follows.

(1) Start with two neurons a and b at random positions w_a and w_b in R^d.

(2) Generate a random input signal ξ according to a density function $P(\xi)$.

(3) Find the nearest neuron (winner neuron) s_1 and the second nearest s_2.

(4) Increase the age of all the edges emanating from s_1.

(5) Add the squared distance between the input signal and the winner neuron to a counter error of s_1:

$$\Delta\text{error}\left(s_1\right) = \left\|w_{s_1} - \xi\right\|^2. \tag{1}$$

(6) Move the winner neuron s_1 and its topological neighbours (neurons connected to s_1) towards ξ by learning steps ε_w and ε_n, respectively, of the total distance:

$$\begin{aligned} \Delta w_{s_1} &= \varepsilon_w \left(\xi - w_{s_1}\right), \\ \Delta w_{s_n} &= \varepsilon_n \left(\xi - w_{s_n}\right). \end{aligned} \tag{2}$$

(7) If s_1 and s_2 are connected by an edge, set the age of this edge to 0. If it does not exist, create it.

(8) Remove the edges larger than $a_{\max}$. If this results in isolated neurons (without emanating edges), remove them as well.

(9) With every certain number λ of input signals generated, insert a new neuron as follows.

(i) Determine the neuron q with the maximum accumulated error.

(ii) Insert a new neuron r between q and its further neighbor f:

$$w_r = 0.5\left(w_q + w_f\right). \tag{3}$$

(iii) Insert new edges connecting the neuron r with neurons q and f, removing the old edge between q and f.

(iv) Decrease the error variables of neurons q and f multiplying them with a constant α. Initialize the error variable of r with the new value of the error variable of q and f.

(10) Decrease all error variables by multiplying them with a constant β.

(11) If the stopping criterion is not yet achieved, go to step 2.

In summary, the adaptation of the network to the input space takes place in step 6. The insertion of connections (step 7) between the two closest neurons to the input patterns establishes an induced Delaunay triangulation in the input space. The elimination of connections (step 8) eliminates the edges that no longer comprise the triangulation. This is made by eliminating the connections between neurons that are no longer activated or isolated. Finally, the accumulated error (step 5) allows the identification of those areas in the input space where it is necessary to increase the number of neurons to improve the mapping.

Using a Growing Neural Gas model to represent 3D data has some advantages over the traditionally used methods like voxel grid or Octrees. For example, we specify the number of neurons (representative points of the map), while other methods like the voxel grid or Octree get different number of occupied cells depending on the distribution and resolution of the cells (voxels on voxel grid and leaves on Octree based methods).

2.2. Octree Based Method. Most 3D point cloud mapping algorithms typically use the spatial organization of the points to encode them in a structure like an Octree to reduce the amount of information. An Octree is a tree data structure in which their internal nodes have exactly eight children. Octrees make a partition of the three-dimensional space by recursively subdividing it into eight octants. It starts from a user specified volume space or it computes the bounding box of the input set. Then, each node or cell is subdivided into 8 children nodes until a certain condition is reached. These conditions vary depending on the problem or the Octree implementation. A commonly used condition is to stop producing new children nodes when the volume or size of the corresponding cell node reaches the desired precision.

One of the main features of the Octree representation is that nodes not containing input space points are not subdivided and therefore those leaf nodes represent an empty volume of the space. This feature is useful for some mobile applications as robot navigation. There exist different approaches to select the representative point of the occupied nodes. A simple one is to get the center of the node cell but using the mean or centroid of the cell inner points improves the preservation of the topology. This approach offers better results but it has a higher computational and memory cost.

2.3. Voxel Grid Method. The VG downsampling technique is based on the input space sampling using a grid of 3D voxels [16]. VG algorithm defines a voxel grid in the 3D space and for each voxel a point is chosen as the representative of all points that lie on that voxel. It is necessary to define the size of the voxels as this size establishes the resolution of the filtered point cloud and therefore the number of points that form the new point cloud. The representative of each cell could be chosen by using one of the approaches described in the

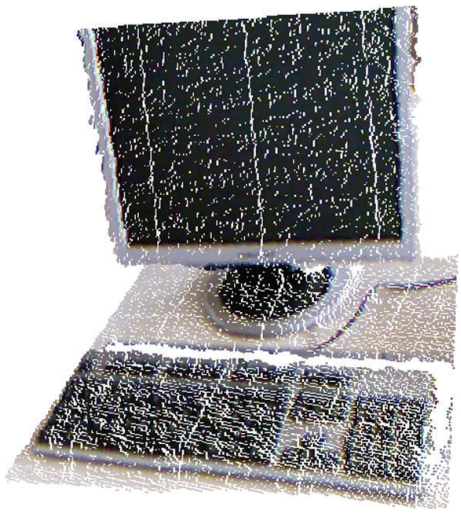


Figure 1: Partial point cloud example.

previous section. Thus, a subset of the input space is obtained that roughly represents the underlying surface.

The VG method, as the Octree based methods, presents the same problems compared to other subsampling techniques: it is not possible to define the final number of points which represents the surface, geometric information loss due to the reduction of the points within a voxel and sensitivity to noisy input spaces.

2.4. Discussion. In this subsection we briefly describe the main differences of the above-described methods. Figures 2, 3 and 4 show several examples of using the three described methods. The GNG representation provides a set of neurons and their neighbors. These representatives and their connections can be used in some algorithms like 3D mesh reconstruction or feature extraction.

Both, voxel grid and Octree methods should provide similar results due to their final representation of the points. In a point cloud reduction application, the Octree gets their representatives of the leaf nodes and if we use the same resolution as the voxel grid method we get a similar division of the space in cubes or cells of the same dimension. The voxel grid method is the most simple and fastest reduction method but it does not have any of the advantages of the Octree structure or GNG model like neighbor searching facilities.

Figure 5 shows a 2D description of the representative points of the described methods. We observe that the GNG method assigns more neurons on high density input areas (bottom left area) than the voxel grid and Octree methods. We also observe how the GNG is able to eliminate some noisy values like the point near the center in contrast with the representatives used in the VG and Octree methods.

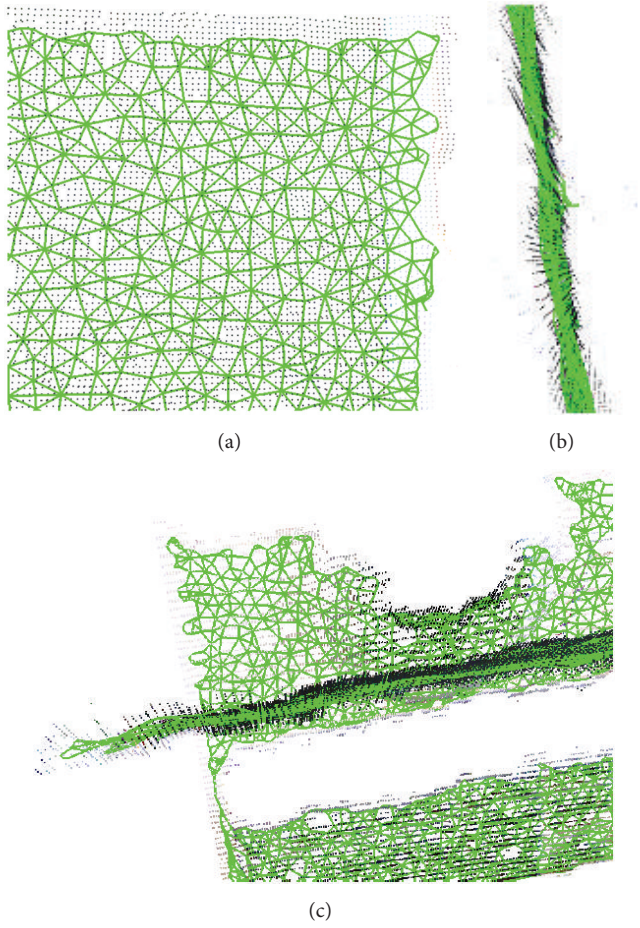


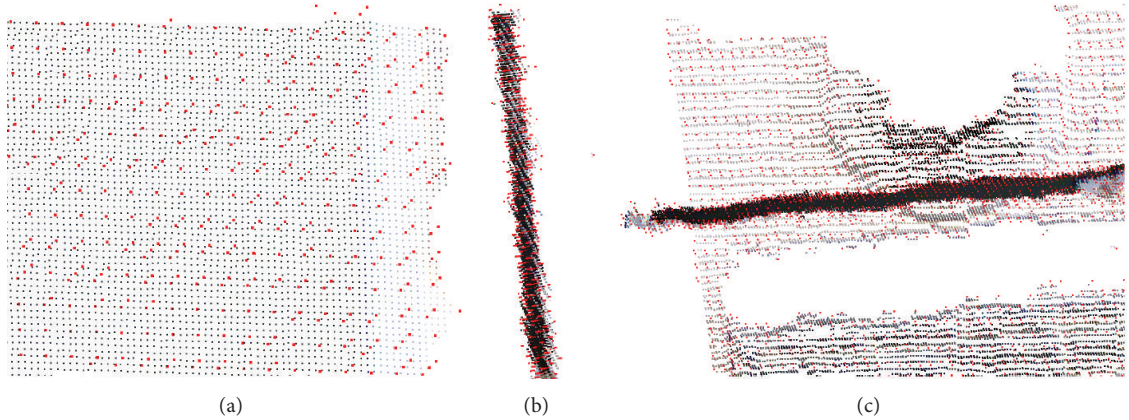
Figure 2: GNG representation example of the partial point cloud shown in Figure 1. Front, side, and top views are shown from (a) to (c), respectively.

3. Experimentation

In this section we test the quality of adaptation of the three described methods. We first describe the data used in the experiments and then we analyze the results of the tested methods, both quantitatively and qualitatively.

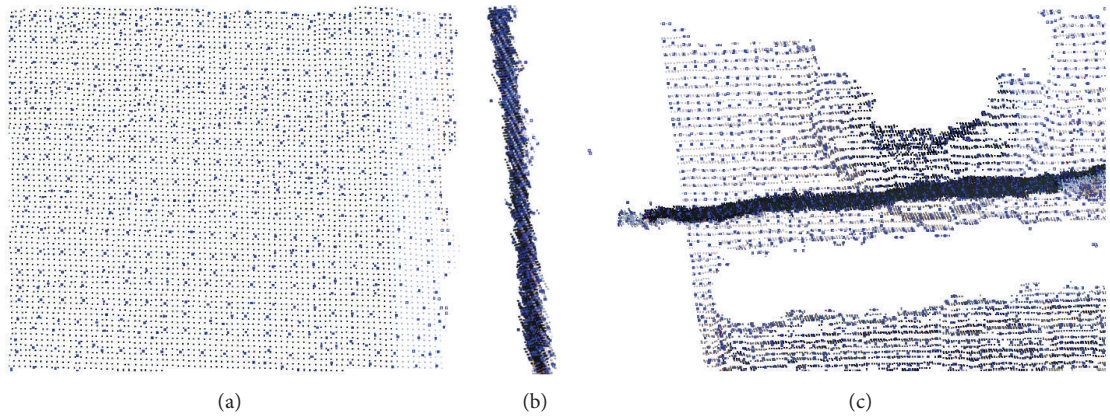
3.1. Experimentation Setup. To test the implemented scene mapping systems on room map scenarios, we used the TUM RGB-D dataset [17]. This dataset provides RGB-D and ground-truth data with the goal of evaluating visual odometry and visual SLAM systems. The dataset contains the color and depth images obtained using a Microsoft Kinect sensor along the ground-truth trajectory of the sensor. It provides images at full frame rate (30 Hz) and sensor resolution (640 × 480). The ground-truth trajectory was obtained from a high-accuracy motion-capture system with eight high-speed tracking cameras (100 Hz).

This dataset contains 39 sequences recorded in two different scenarios. The fr1 datasets were recorded in a typical office environment (first scenario) and the fr2 datasets were



(a) (b) (c)

Figure 3: Octree representation example of the partial point cloud shown in Figure 1. Front, side, and top views are shown from (a) to (c), respectively.



(a) (b) (c)

Figure 4: Voxel grid representation example of the partial point cloud shown in Figure 1. Front, side, and top views are shown from (a) to (c), respectively.

Table 1: Number of points of each ground-truth map dataset.

Dataset	Number of input points
fr1 *xyz*	1049739
fr1 desk	1952544
fr1 360	2357039
fr1 desk2	2751402
fr2 *xyz*	3492032
fr2 desk	5841800
fr3 long	1636623

recorded in a large industrial hall (second scenario). Figures 6 and 7 show the ground-truth reconstruction maps of the "fr1 360" and "fr2 desk," respectively.

Table 1 shows the number of points of the input maps used in the experimentation. We can observe that the number of input points ranges from one million ("fr1 *xyz*") to 6 million ("fr2 desk").

3.2. Quality Adaptation Experiment. As we previously mentioned, we are going to compare the proposed GNG adaptation with two commonly used data structures in the state-of-the-art, Octree, and voxel grid. The implementation of both methods is included in the Point Cloud Library (PCL) (the Point Cloud Library (or PCL) is a large scale, open project [18] for 2D/3D image and point cloud processing). The Octree implementation uses the center of the leaf nodes as representative points. However, the voxel grid implementation uses the centroid of the points of each nonempty voxel. Both implementations use a resolution parameter that represents the size of the voxel in the VG method and the side of the

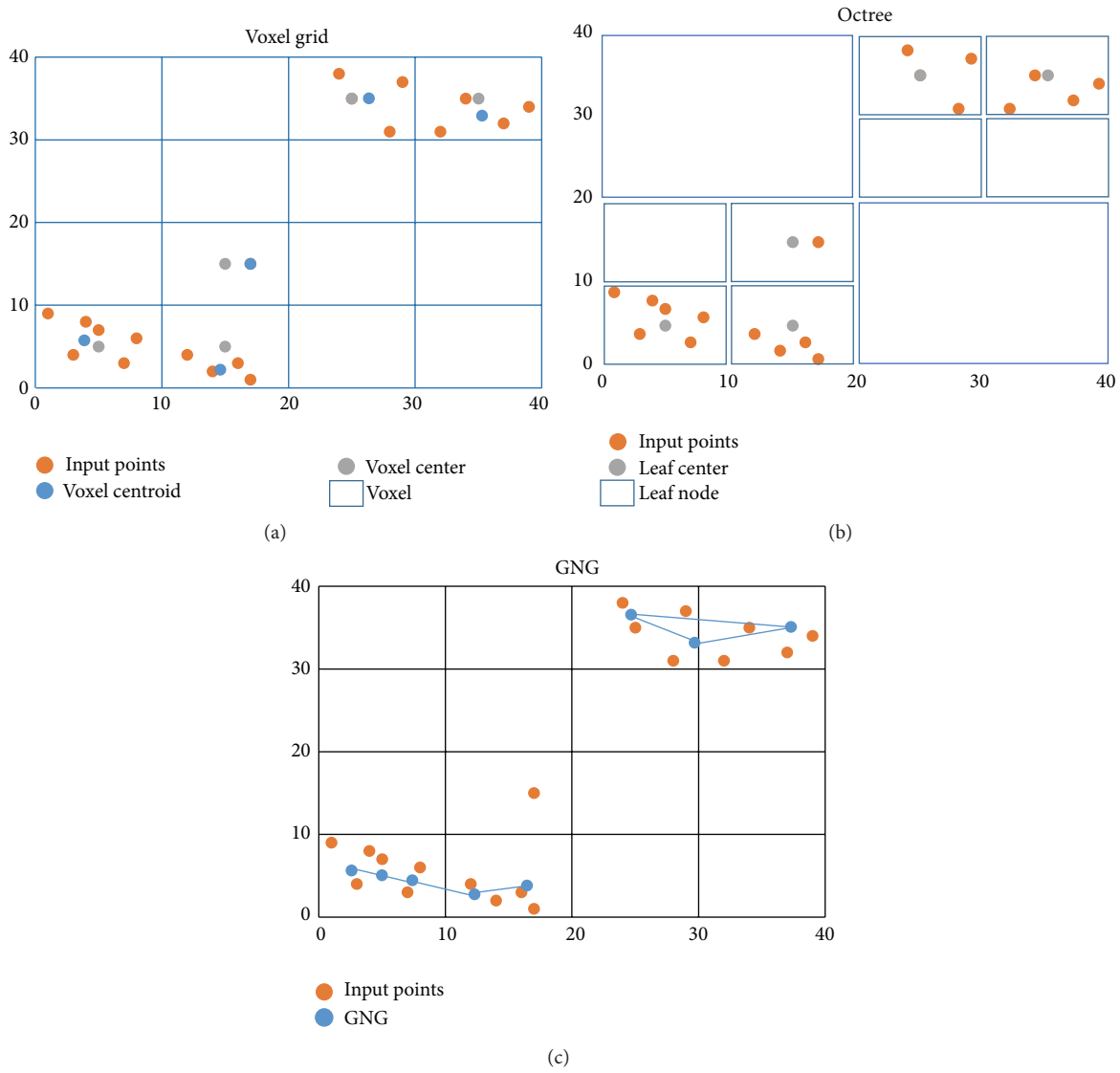


Figure 5: Two-dimensional examples of the three tested methods.

Figure 6: Example of the "fr1 360" ground-truth point cloud map.

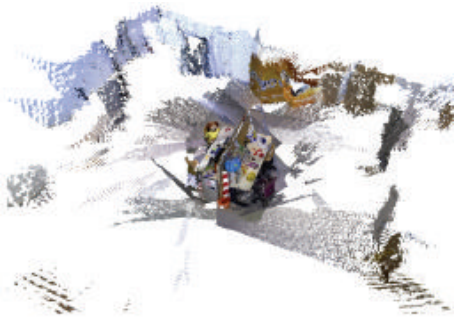


Figure 7: Example of the "fr3 long" ground-truth point cloud map.

leaf cell of the Octree implementation. The GNG results are obtained using 10,000 λ input patterns.

We extensively tested the implemented methods using different number of representatives. Since the three tested methods reduce the amount of noise in the generated map, it is needed to know the real distance from the selected representatives to the original input space. The following measure specifies how close the representations are from the original model.

A quantitative measure of the input space adaptation of the generated map is obtained by computing the mean error (ME) of the reduced map against sampled points (input space):

$$\mathrm{ME} = \frac{1}{|V|} \sum_{p \in V} \min_{q \in A} \|p - q\|, \tag{4}$$

where V is the input space, p is a point that belongs to the input space, and q is the representative point with the minimum distance to p. Euclidean distances to closest points are averaged over the entire input space.

Figure 8 shows the RMS errors of the three methods on the six different tested maps. We observe that the three methods have similar behavior on the different datasets. The Octree method gets the worse results probably due to the selection of its cell-node center as representative. The voxel grid gets lower errors than Octree due to the use of the centroid of the inner points instead of using the center of the voxel or cell.

It is important to point out again that the representative selection method used in this comparison is given by the used implementations. But Octree and voxel grid methods can use a representative selection strategy. GNG adaption shows the best results on all datasets. It is noticeable that the GNG gets lower errors with different number of representatives but as the number of representatives increases the three different methods converge to the same error.

3.3. Qualitative Results. In this subsection we qualitatively analyze the results obtained with the three different methods. Figure 9 shows the original map and the three tested representations of the "fr1 360" scene. Part (a) shows the point cloud that we are trying to represent and reduce. Parts (b) and (c) are, respectively, the Octree and voxel grid representation, and part (d) is the GNG representation of the scene. The Octree representation, using the centers of the leaf nodes, gets a strongly structured point representation. This representation gets a more uniform distribution of the representatives but the adaption error is worse as we saw in Figure 5 and the mean error graphs. The voxel grid representation gets similar results compared to the Octree, where the points are uniformly distributed as it can be observed in the points that represent the floor. However, it gets better results on the boundaries compared with the Octree method. Both VG and Octree place representatives in isolated and noisy points. However, the GNG neurons are uniformly distributed over the input space and the impact of the noisy points and undefined borders on the obtained representation is reduced. We also observe the inherent triangulation of the space that the GNG algorithm gets using the neurons neighborhood.

Figure 10 shows two more experiments with GNG representation. In these experiments we have selected two maps with holes. GNG is able to adapt to these holes and it does not insert neurons in them.

Table 2: Runtimes and speed-up of GPU versus CPU implementation for different GNG versions.

Neurons	Patterns	CPU runtime (s)	GPU speed-up	GPU runtime (s)
5000	250	63	3×	21
12000	350	526	5×	105.2
18000	500	1448	6×	241.33

3.4. Execution Time. With respect to computational cost, our method is feasible to be included in a modern system using general purpose computing platforms. However, we designed in a previous work [17] a GPU-based implementation of the GNG algorithm that speeds up the sequential version several times. The speedup is increased as the number of neurons used for the representation grows.

In Table 2 some results can be appreciated with different number of neurons and input patterns with CPU and GPU runtimes and speedup obtained with GPU version with respect to the CPU ones. The GPU used was a GTX 480 NVIDIA graphic card with 480 cores, a global memory of 1.5 MB, and a bandwidth memory of 177.4 GB/sec.

3.5. GNG Maps Applications. The experiments showed how the GNG is able to adapt their topology to represent the input map space. In [19] we showed many applications of the GNG structure in order to improve 3D data representation and computer vision methods. We proved that the GNG representation partially discards most of the noisy values provided by these RGB-D sensors. In addition, we performed some experiments to show the improvement obtained using

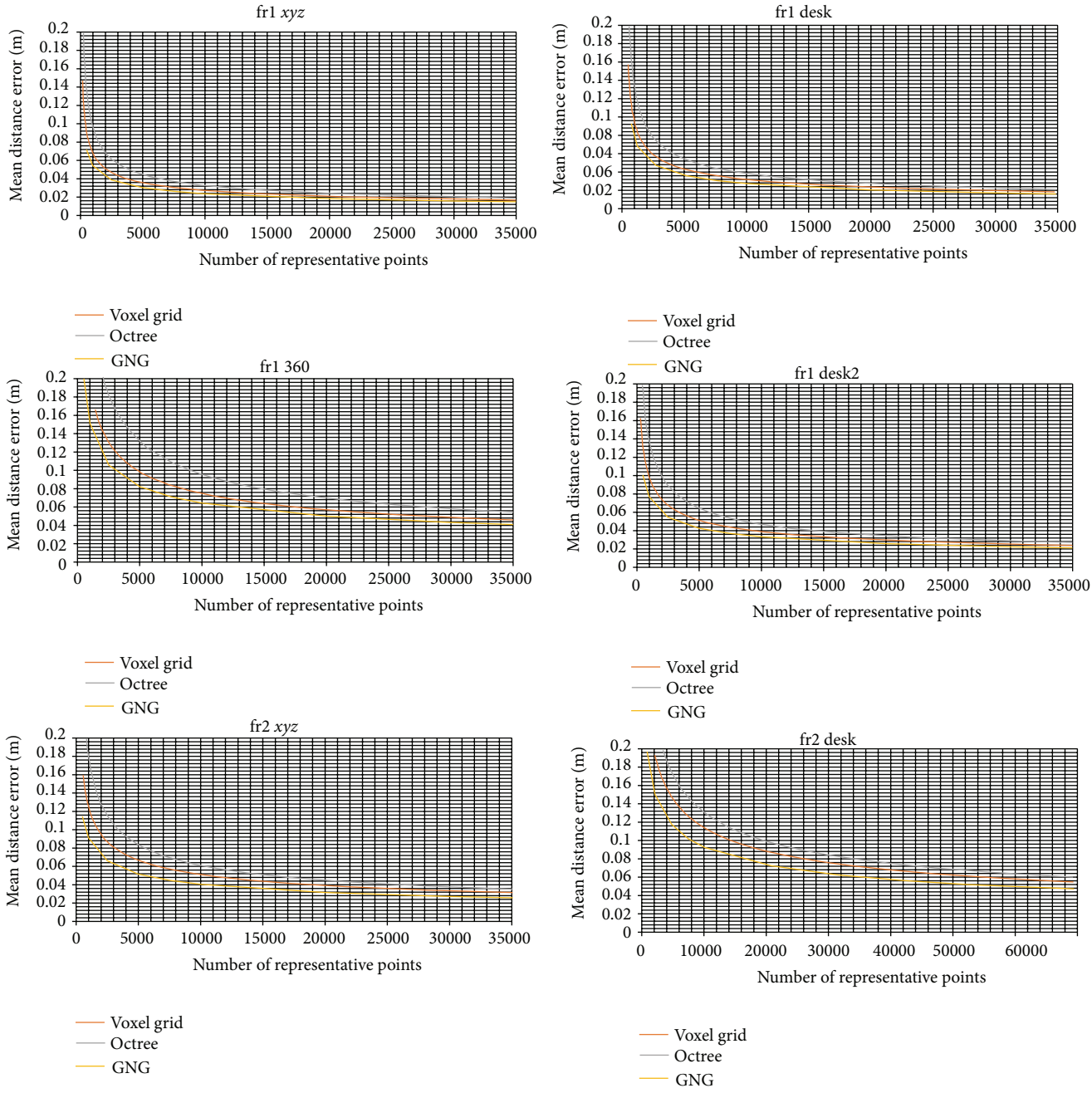
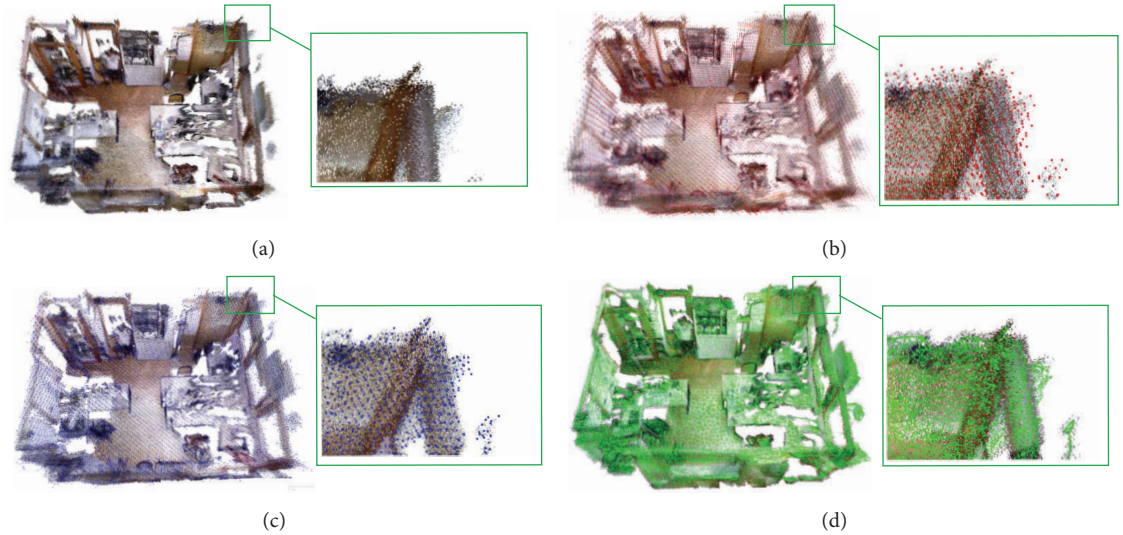


Figure 8: Closest neighbor distance mean errors of the different datasets.

the GNG representation on normal estimation using point neighborhood and keypoint detection and description.

Normal estimation methods are based on the analysis of the eigenvectors and eigenvalues of a covariance matrix created from the nearest neighbours and are very sensitive to noisy data. Therefore, we computed normals on raw and filtered point clouds in order to demonstrate how a simple 3D processing step like normal or curvature estimation is affected by the presence of noise.

Figure 11 shows how more stable normals are estimated using filtered point clouds produced by the GNG method. Normals are considered more stable as their distribution is smooth and also they have less abrupt changes in their directions. Moreover, a flat wall with some small changes in its surface was selected to appreciate changes in normal directions. We employed 20,000 neurons and 1,000 λ patterns as configuration parameters for the GNG method in the normal estimation experiment Figure 11(b).



(a) (b) (c) (d)

Figure 9: (a) Original point cloud map. (b) Octree reduction. (c) Voxel grid reduction. (d) GNG representation.

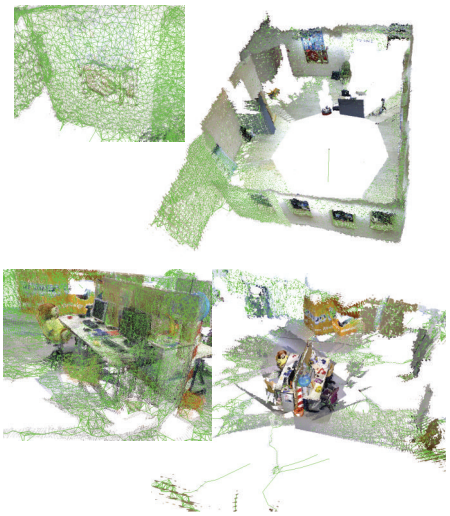
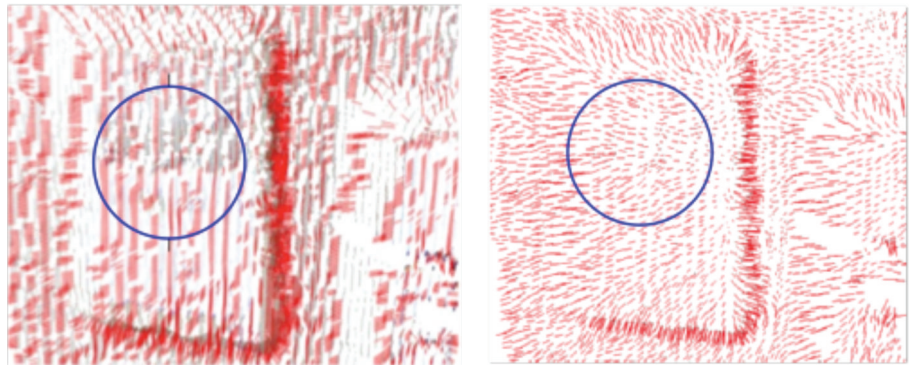


Figure 10: Other examples of GNG representation with two additional maps. Left part is a zoomed detail.



(a) (b)

Figure 11: Normal estimation comparison. (a) Normal estimation on raw point cloud. (b) Normal estimation on filtered point cloud produced by the GNG method.

Table 3: RMS deviation error (meters) is obtained using different detector-descriptor combinations. Combinations are computed on the original point cloud (raw) and different filtered point clouds using voxel grid and the proposed GNG method. Keypoint detector search radius is equal to 0.05 meters. Feature extractor search radius is equal to 0.02 meters.

		GNG			VG			Raw
		$5000p$	$10000p$	$20000p$	$5000p$	$10000p$	$20000p$	All points
SIFT3D	FPFH	0.168	0.092	0.231	0.239	0.073	0.139	0.103
	CSHOT	0.052	0.037	0.019	0.063	0.07	0.037	0.039
	PFH	0.185	0.367	0.255	0.54	0.171	0.375	0.082
	PFHRGB	0.106	0.029	0.057	0.08	0.05	0.027	0.041
Harris3D	FPFH	0.151	0.114	0.079	0.404	0.088	0.18	0.128
	CSHOT	0.085	0.046	0.052	0.038	0.033	0.069	0.066
	PFH	0.109	0.177	0.153	0.305	0.097	0.469	0.144
	PFHRGB	0.054	0.047	0.042	0.033	0.058	0.043	0.093
Tomasi3D	FPFH	0.049	0.054	0.27	0.383	0.127	0.148	0.659
	CSHOT	0.043	0.023	0.063	0.022	0.047	0.046	0.049
	PFH	0.189	0.308	0.319	0.067	0.258	0.123	0.765
	PFHRGB	0.112	0.121	0.066	0.066	0.049	0.078	0.062
Noble3D	FPFH	0.143	0.186	0.199	0.387	0.188	0.289	0.114
	CSHOT	0.098	0.052	0.063	0.085	0.059	0.048	0.06
	PFH	0.273	0.127	0.239	0.073	0.244	0.188	0.099
	PFHRGB	0.188	0.117	0.064	0.096	0.082	0.079	0.077
Lowe3D	FPFH	0.143	0.117	0.076	0.387	0.188	0.093	0.203
	CSHOT	0.065	0.052	0.062	0.04	0.059	0.107	0.067
	PFH	0.273	0.12	0.239	0.073	0.244	0.167	0.24
	PFHRGB	0.188	0.135	0.016	0.077	0.082	0.076	0.027
Curvature3D	FPFH	0.099	0.228	0.113	0.228	0.103	0.093	—
	CSHOT	0.048	0.032	0.022	0.033	0.042	0.05	—
	PFH	0.262	0.151	0.123	0.244	0.101	0.057	—
	PFHRGB	0.139	0.071	0.053	0.068	0.083	0.023	—

In order to test the keypoint detector/descriptor improvement, we performed a transformation estimation algorithm and we compared the GNG results against the voxel grid representation and against the entire source point cloud. We used the available descriptors [18, 20, 21] and detectors [22] provided by the Point Cloud Library [23]. The minimum, median, mean, and maximum of RMS transformation error (with respect to different keypoint detectors) are presented in Table 3. These results show how filtered point clouds using the GNG method generally improved the precision of the estimated transformation. Moreover, the worst estimated transformations (maximum errors) were also slightly improved using the GNG compared to the other techniques.

4. Conclusions

3D maps obtained from RGB-D data are useful for robotics tasks, like robot navigation. But this kind of maps contains a huge amount of data, which must be reduced to properly process the map. In this paper, we have presented a method to represent and reduce 3D maps. Our method is based on a GNG neural network which is adapted to the 3D input space. The experiments carried out demonstrated the validity of our method, as it provided better adaptation than two of the most used methods for these tasks: voxel grid and Octree.

As future works, we propose to extend our method to provide a useful map for robot navigation. We also plan to provide the GNG with a way to revert the reduction or compression of the points, storing information in the neurons neighborhood (color, point distribution, etc.).

Conflict of Interests

The authors declare that there is no conflict of interests regarding the publication of this paper.

Acknowledgments

This work was partially funded by the Spanish Government DPI2013-40534-R grant. Experiments were made possible with a generous donation of hardware from NVDIA.

References

[1] D. Fox, S. Thrun, and W. Burgard, *Probabilistic Robotics*, The MIT Press, Cambridge, Mass, USA, 2005.

[2] M. Herbert, C. Caillas, E. Krotkov, I. S. Kweon, and T. Kanade, "Terrain mapping for a roving planetary explorer," in *Proceedings of the IEEE International Conference on Robotics and Automation*, vol. 2, pp. 997–1002, 1989.

[3] R. Triebel, P. Pfaff, and W. Burgard, "Multi-level surface maps for outdoor terrain mapping and loop closing," in *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS '06)*, pp. 2276–2282, October 2006.

[4] A. Elfes, "Using occupancy grids for mobile robot perception and navigation," *Computer*, vol. 22, no. 6, pp. 46–57, 1989.

[5] P. Stepan, M. Kulich, and L. Preucil, "Robust data fusion with occupancy grid," *IEEE Transactions on Systems, Man, and Cybernetics C*, vol. 35, no. 1, pp. 106–115, 2005.

[6] B. Schiele and J. L. Crowley, "A comparison of position estimation techniques using occupancy grids," *Robotics and Autonomous Systems*, vol. 12, no. 3-4, pp. 163–171, 1994.

[7] C. I. Connolly, "Cumulative generation of octree models from range data," in *Proceedings of the 1st International Conference on Robotics and Automation*, vol. 1, pp. 25–32, March 1984.

[8] K. M. Wurm, A. Hornung, M. Bennewitz, C. Stachniss, and W. Burgard, "OctoMap: a probabilistic, flexible, and compact 3D map representation for robotic systems," in *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA '10)*, 2010.

[9] L. Kobbelt and M. Botsch, "A survey of point-based techniques in computer graphics," *Computers and Graphics*, vol. 28, no. 6, pp. 801–814, 2004.

[10] L. Wang, J. Chen, and B. Yuan, "Simplified representation for 3D point cloud data," in *Proceedings of the IEEE 10th International Conference on Signal Processing (ICSP '10)*, pp. 1271–1274, October 2010.

[11] D. Viejo, J. Garcia, M. Cazorla, D. Gil, and M. Johnsson, "Using GNG to improve 3D feature extraction: application to 6DoF egomotion," *Neural Networks*, vol. 32, pp. 138–146, 2012.

[12] T. Kohonen, *Self-Organizing Maps*, vol. 30 of *Springer Series in Information Sciences*, Springer, Berlin, Germany, 1995.

[13] B. Fritzke, "A growing neural gas network learns topologies," in *Advances in Neural Information Processing Systems*, vol. 7, pp. 625–632, MIT Press, 1995.

[14] T. Martinetz, "Competitive hebbian learning rule forms perfectly topology preserving maps," in *Proceedings of the International Conference on Artificial Neural Networks (ICANN '93)*, pp. 427–434, 1993.

[15] T. Martinetz and K. Schulten, "Topology representing networks," *Neural Networks*, vol. 7, no. 3, pp. 507–522, 1994.

[16] G. Xu, B. Mourrain, R. Duvigneau, and A. Galligo, "Analysis-suitable volume parameterization of multi-block computational domain in isogeometric applications," *Computer-Aided Design*, vol. 45, no. 2, pp. 395–404, 2013.

[17] S. Orts, J. Garcia-Rodriguez, D. Viejo, M. Cazorla, and V. Morell, "GPGPU implementation of growing neural gas: Application to 3D scene reconstruction," *Journal of Parallel and Distributed Computing*, vol. 72, no. 10, pp. 1361–1372, 2012.

[18] F. Tombari, S. Salti, and L. Di Stefano, "A combined texture-shape descriptor for enhanced 3D feature matching," in *Proceedings of the 18th IEEE International Conference on Image Processing (ICIP '11)*, pp. 809–812, September 2011.

[19] S. Orts-Escolano, *A Three-Dimensional Representation Method for Noisy Point Clouds Based on Growing Self-Organizing Maps Accelerated on GPUs*, University of Alicante, 2014.

[20] R. B. Rusu, N. Blodow, Z. C. Marton, and M. Beetz, "Aligning point cloud views using persistent feature histograms," in *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS 08)*, pp. 3384–3391, Nice, France, September 2008.

[21] R. B. Rusu, N. Blodow, and M. Beetz, "Fast point feature histograms (FPFH) for 3D registration," in *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA '09)*, pp. 3212–3217, Kobe, Japan, May 2009.

[22] I. Sipiran and B. Bustos, "Harris 3D: a robust extension of the Harris operator for interest point detection on 3D meshes," *Visual Computer*, vol. 27, no. 11, pp. 963–976, 2011.

[23] R. B. Rusu and S. Cousins, "3D is here: Point Cloud Library (PCL)," in *Proceeding of the IEEE International Conference on Robotics and Automation (ICRA '11)*, pp. 1–4, Shanghai, China, May 2011.

Hindawi Publishing Corporation
Mathematical Problems in Engineering
Volume 2014, Article ID 905826, 11 pages
http://dx.doi.org/10.1155/2014/905826

Research Article

A Bioinspired Neural Model Based Extended Kalman Filter for Robot SLAM

Jianjun Ni,[1,2] Chu Wang,[1] Xinnan Fan,[1] and Simon X. Yang[3]

[1] *College of IOT Engineering, Hohai University, Changzhou 213022, China*
[2] *Changzhou Key Laboratory of Sensor Networks and Environmental Sensing, Hohai University, Changzhou 213022, China*
[3] *Advanced Robotics and Intelligent Systems (ARIS) Laboratory, School of Engineering, University of Guelph, Guelph, ON, Canada N1G 2W1*

Correspondence should be addressed to Jianjun Ni; njjhhuc@gmail.com

Received 15 March 2014; Revised 22 July 2014; Accepted 22 July 2014; Published 13 August 2014

Academic Editor: Yi Chen

Robot simultaneous localization and mapping (SLAM) problem is a very important and challenging issue in the robotic field. The main tasks of SLAM include how to reduce the localization error and the estimated error of the landmarks and improve the robustness and accuracy of the algorithms. The extended Kalman filter (EKF) based method is one of the most popular methods for SLAM. However, the accuracy of the EKF based SLAM algorithm will be reduced when the noise model is inaccurate. To solve this problem, a novel bioinspired neural model based SLAM approach is proposed in this paper. In the proposed approach, an adaptive EKF based SLAM structure is proposed, and a bioinspired neural model is used to adjust the weights of system noise and observation noise adaptively, which can guarantee the stability of the filter and the accuracy of the SLAM algorithm. The proposed approach can deal with the SLAM problem in various situations, for example, the noise is in abnormal conditions. Finally, some simulation experiments are carried out to validate and demonstrate the efficiency of the proposed approach.

1. Introduction

Robot simultaneous localization and mapping (SLAM) is one of the most challenging problems in mobile robotic fields, which has important theory and application value in various robotic applications, such as the underwater detection, domestic service, and universe exploration [1–5]. Various approaches have been proposed to deal with the SLAM problem. Mullane et al. [6] proposed an integrated Bayesian framework for feature-based SLAM in the general case of uncertain feature number and data association. Lui and Jarvis [7] presented a pure vision-based topological SLAM system for mobile robot autonomous navigation in initially unknown environments. Chatterjee and Matsuno [8] proposed a new neurofuzzy based adaptive Kalman filtering algorithm for SLAM of mobile robots or vehicles. Zhou et al. [9] proposed an auxiliary marginal particle filter and FastSLAM based compositive SLAM algorithm to improve the performance of samples and increase the estimation accuracy. Kaess et al. [10] presented a novel approach to the simultaneous localization and mapping problem that is based on fast incremental matrix factorization. Benedettelli et al. [11] proposed a multirobot cooperative SLAM algorithm using M-Space representation of linear features, suitable for environments which can be represented in terms of lines and segments.

Previous research on SLAM problem may be divided into two broad categories. One category consists of a number of mathematical probabilistic techniques, such as EKF based algorithm [12, 13], particle filtering based algorithm [14, 15], and Monte Carlo localization method [16, 17]. The other category of research focuses on emulating the biological systems thought to be responsible for mapping and navigation in animals [18–20]. However, both of the category algorithms have some limitations. For example, the standard Kalman filtering is based on the minimum variance estimation; the filter will exhibit a divergence phenomenon with the increase in the number of filterings; namely, the error between the estimated value and the actual value will become bigger and bigger. The biological emulating based algorithms are too

complex to be realized and the SLAM mechanism of animals is not very clear now.

Although there are many modern techniques that are suitable for SLAM, the EKF based SLAM method is still one of the most popular methods for SLAM, because the EKF based method has a good algorithm structure for SLAM and a strict mathematical theory basis. The EKF based method is used widely to solve SLAM problem [8, 21–24], especially in some applications, where the computational constraints or the scarcity of high precision sensors makes it impossible to use other SLAM methods, such as the methods based on high-resolution vision sensors [25–27] and the methods based on high dense topological maps [28, 29]. However, there is a key problem of the EKF based SLAM method; namely, the accuracy of the system noise and the observation noise model will decide the final accuracy of the SLAM algorithm. To deal with this problem, some improved algorithms have been proposed. For example, Du et al. [30] proposed an improved fuzzy adaptive EKF to establish a priori noise model for the SLAM problem. Kang et al. [31] presented a modified neural network aided EKF based SLAM for improving the accuracy of the feature map. However, those algorithms introduced above still have some limitations, such as the computation which is complex and the algorithm is unstable in some conditions.

To improve the robustness and accuracy of the EKF based SLAM algorithm, a novel adaptive EKF based SLAM approach is proposed in this paper. Firstly, an adaptive EKF based SLAM algorithm structure is proposed, where the weights of noises in the EKF based SLAM algorithm are adjusted adaptively. Secondly, a bioinspired neural model is used to realize these adjustments for noise weights, which can guarantee the stability of the filter. Finally, some simulation experiments were conducted. The results show the efficiency of the proposed approach. The divergence problem of EKF is solved effectively, and the robustness and accuracy of the SLAM approach are improved.

This paper is organized as follows. Section 2 presents the proposed bioinspired neural model based extended Kalman filter algorithm. The simulation experiments for mobile robot SLAM task are given in Section 3. Section 4 discusses the sensitivity of the parameters and some performances of the proposed approach in detail. Finally, the conclusions are given in Section 5.

2. The Proposed Bioinspired Neural Model Based EKF Algorithm

In this paper, the SLAM problem is studied. The core task of SLAM is that a robot explores in an unknown environment to learn the environment (map), while simultaneously using that map to locate within the environment [32–34]. A novel approach based on bioinspired neural model is proposed. In the proposed approach, EKF is used to solve the SLAM problem, while a bioinspired neural model is used to adjust the weights of system noise and observation noise adaptively in the EKF based SLAM algorithm. The main reason to use this bioinspired neural model is that it is not only a feasible solution, but also an efficient one. The bioinspired method can deal with SLAM problem efficiently without any a priori knowledge nor any learning procedures, which is a trend in the SLAM field [32, 35]. The basic work flow and the theoretical analysis of the EKF based SLAM algorithm can be viewed in [8, 36, 37]. The proposed approach is introduced in detail as follows.

2.1. The SLAM Algorithm Based on EKF. The basic idea of EKF based SLAM is that the robot uses the known states to predict the states of the next step, and then this prediction is corrected based on the observation at the next step. The state equation of EKF based algorithm is as follows [38]:

$$\begin{aligned} X(k+1) &= F(k)X(k) + G(k)U(k) + W(k), \\ Z(k) &= H(k)X(k) + V(k), \end{aligned} \tag{1}$$

where $X(k)$ is the state variable of the system at time k; $U(k)$ is the control variable of the system; $Z(k)$ is the observation value by robotic sensors; $F(k)$ is the state transition matrix; $G(k)$ is the control matrix; $H(k)$ is the observation matrix; $W(k)$ and $V(k)$ are the noise of system and observation, respectively. Here the state of the system is based on the specific SLAM tasks, such as the number of landmarks and the dimension of the environment. Assume that the location of the robot at time k is $x_v(k)$ and the position of the ith landmark is l_i; then the state of the system can be denoted by

$$X(k) = \left[x_v^T(k), l_1^T, l_2^T, \ldots, l_N^T\right]^T, \tag{2}$$

where N is the number of landmarks in the environment. The location of the robot $x_v(k)$ is updated by its kinematic model, while the landmarks in the environment are assumed as static, namely, $l_i(k+1) = l_i(k) = l_i$. The general EKF based SLAM algorithm includes two main steps.

(1) Prediction Step. Firstly, the state equation is used to predict the state of system at the next time. Assume that the system state calculated at time k is $X(k \mid k)$; the prediction of system state for the next time $k+1$ is denoted by $X(k+1 \mid k)$, which can be predicted as follows:

$$\begin{aligned} X(k+1 \mid k) &= F(k)X(k \mid k) + G(k)U(k), \\ Z(k+1 \mid k) &= H(k)X(k+1 \mid k), \\ P(k+1 \mid k) &= F(k)P(k \mid k)F^T(k) + Q(k), \end{aligned} \tag{3}$$

where $P(k+1 \mid k)$ and $P(k \mid k)$ are the covariance matrix related to $X(k+1 \mid k)$ and $X(k \mid k)$, respectively; and $Q(k)$ is the covariance matrix of control noise.

(2) Update Step. When the observation of system at time $k+1$ is obtained by the onboard sensors of the robot, the system state at time $k+1$ can be updated based on the system state equation and the prediction of system state:

$$\begin{aligned} &X(k+1 \mid k+1) \\ &\quad = X(k+1 \mid k) + K(k+1)\left(Z(k+1) - Z(k+1 \mid k)\right), \\ &P(k+1 \mid k+1) \\ &\quad = P(k+1 \mid k) - K(k+1)H(k+1)P(k+1 \mid k), \end{aligned} \tag{4}$$

where $K(k)$ is the gain of EKF, which can be obtained by

$$K(k+1) = \frac{P(k+1 \mid k) H^T(k+1)}{H(k+1) P(k+1 \mid k) H^T(k+1) + R(k+1)}, \tag{5}$$

where $R(k)$ is the covariance matrix of observation noise.

In this paper, the difference between the prediction of system state and the real value obtained by the robotic onboard sensors is defined as the innovation, which is denoted by $C(k)$. The innovation $C(k)$ can be calculated by

$$C(k+1) = Z(k+1) - Z(k+1 \mid k). \tag{6}$$

The variance of the innovation $C(k)$ is defined as $S(k)$, which can be calculated by

$$S(k+1) = H(k+1) P(k+1 \mid k) H^T(k+1) + R(k+1). \tag{7}$$

The innovation $C(k)$ can be used to adjust and correct the EKF based method. In this paper, an adaptive EKF based SLAM algorithm is proposed, where a bioinspired neural model is used to adjust the credibility and availability of the observation.

2.2. The Proposed Adaptive Model for the EKF Based SLAM. Before the introduction of the bioinspired neural model based EKF algorithm, it is necessary to set up a control model for the EKF based SLAM firstly [30]. In this model, the noises in the EKF based SLAM algorithm are expressed as follows:

$$\begin{aligned} Q(k) &= a(k) Q, \\ R(k) &= b(k) R, \end{aligned} \tag{8}$$

where Q and R are the initial covariance matrixes for noises; $a(k)$ and $b(k)$ are the weights for system noise and observation noise, respectively.

In a certain time window M, the mean values of the innovation $C(k)$ and its variance $S(k)$ are denoted by $\overline{C}(k)$ and $\overline{S}(k)$, respectively, which can be calculated as follows:

$$\begin{aligned} \overline{C}(k) &= \frac{1}{M} \sum_{i=k-M+1}^{k} C(i), \\ \overline{S}(k) &= \frac{1}{M} \sum_{i=k-M+1}^{k} S(i) S^T(i). \end{aligned} \tag{9}$$

The confidence levels of the mean values of the innovation $C(k)$ and its variance $S(k)$ are denoted by q_1 and q_2, respectively, which can be obtained by

$$\begin{aligned} q_1(k) &= \frac{(1/J) \sum_{j=1}^{J} \overline{C}_j(k)}{Z_j(k)} \\ q_2(k) &= \frac{\mathrm{Tr}\left(\overline{S}(k)\right)}{\mathrm{Tr}(S(k))}, \end{aligned} \tag{10}$$

where J is the dimension of the observation and the Tr() is a function to calculate the trace of a matrix. In the EKF model, q_1 and q_2 will fluctuate around 0 and 1 in a very little range, respectively, if the assumption of the noise model is accurate. Otherwise, the values of $\overline{C}(k)$ and $\overline{S}(k)$ will increase when the real value of the observation noise increases. Then the value of q_1 will be far away from 0 and the value of q_2 will be bigger than 1. Under this condition, the stability of the system will become worse [30, 36, 38].

Remark 1. Of course, if the parameters of the noise model are properly tuned and assuming correspondences are known, the covariance of the noise will become arbitrarily small and the accuracy of the EKF based SLAM algorithm can be guaranteed. But it needs more experience and time, which are both scarce in the real-world applications of SLAM tasks.

To deal with the problem discussed above, an adaptive model is proposed in this paper. The basic idea of this model is to change the weights of the noises if the assumption of the noise model is away from the actual value. The adaptive EKF based SLAM algorithm structure is shown in Figure 1. In this paper, a bioinspired neural model is used to realize the adjustment function.

2.3. A Bioinspired Neural Model for the EKF Based SLAM. In the proposed approach, a bioinspired neural dynamic model is used to adjust the weights of noises adaptively. The first computational model of a uniform patch of membrane in a biological neural system using electrical circuit elements was proposed by Hodgkin and Huxley [39, 40]. In their membrane model, the dynamics of the voltage across the membrane is described using a state equation technique (see [39, 41]). Based on this state equation, a shunting model is obtained as

$$\frac{dx_i}{dt} = -Ax_i + \left(B - x_i\right) S_i^+ - \left(D + x_i\right) S_i^-, \tag{11}$$

where x_i is the neural activity (membrane potential) of the ith neuron; A, B, and D are nonnegative constants, representing the passive decay rate, and the upper and lower bounds of the neural activity, respectively; and S_i^+ and S_i^- are the excitatory and inhibitory inputs to the neuron. This shunting model was first proposed by Grossberg to understand the real-time adaptive behavior of individuals to complex and dynamic environmental contingencies and has many applications in visual perception, sensory motor control, and many other areas (e.g., [41, 42]). This shunting model is stable [39, 43], where the neural activity x_i is bounded in the finite interval $[-D, B]$. The state workspace varies in terms of the dynamics of the neural model, due to the influence of the external inputs.

In the proposed EKF based SLAM algorithm structure (see Figure 1), the prediction and update steps are carried out continuously. As a result, the confidence levels of the mean values of the innovation $C(k)$ and its variance $S(k)$ will change constantly. Because the system tends to be stable if q_1 is close to 0 and q_2 is close to 1, the weights $a(k)$ and $b(k)$ of the system

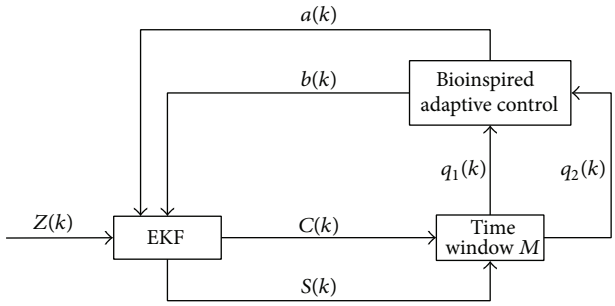


Figure 1: The adaptive EKF based SLAM algorithm structure based on the bioinspired neural model.

noise and observation noise in the EKF based SLAM algorithm should be controlled in real time. As introduced above, the bioinspired neural model has several advantages such as the guaranteed stability and efficient computation [41, 44], so it could be used to improve the adaptability of the EKF based SLAM algorithm. The basic idea of the proposed approach is to realize the control function based on the bioinspired neural model above. The inputs of this controller are $q_1(k)$ and $q_2(k)$, and the outputs of this controller are $a(k)$ and $b(k)$. The adaptive control function based on this bioinspired neural model (namely, the shunting model above; see (11)) is defined as

$$\begin{aligned}\frac{da(k)}{dt} &= -A_1 a(k) + \left(B_1 - a(k)\right) f_1 - \left(D_1 + a(k)\right) f_2,\\ \frac{db(k)}{dt} &= -A_2 b(k) + \left(B_2 - b(k)\right) f_1 - \left(D_2 + b(k)\right) f_2,\end{aligned} \tag{12}$$

where A_1, A_2, B_1, B_2, D_1, and D_2 are the parameters of the controller for $a(k)$ and $b(k)$; f_1 and f_2 are the inputs of the bioinspired neural model, which are defined as

$$\begin{aligned}f_1 &= q_1(k),\\ f_2 &= 1 - q_2(k).\end{aligned} \tag{13}$$

Remark 2. Based on the proposed bioinspired model, the weights for system noise and observation noise (namely, $a(k)$ and $b(k)$) can be guaranteed in a stable range and change with the difference between the prediction of the system state and the real value obtained by the robotic onboard sensors (namely, $q_1(k)$ and $q_2(k)$) adaptively. In addition, this bioinspired neural model based controller does not need any a priori knowledge about the noise model nor any learning procedures.

3. Simulation Experiments

In this paper, to test the performance of the proposed approach, some simulation experiments are conducted which were coded in MATLAB. In these experiments, a robot with some sensors moves in a predetermined trajectory and some SLAM algorithms are used to map the environment and keep track of the robot position simultaneously (see Figure 2). In these experiments, the noise model is unknown.

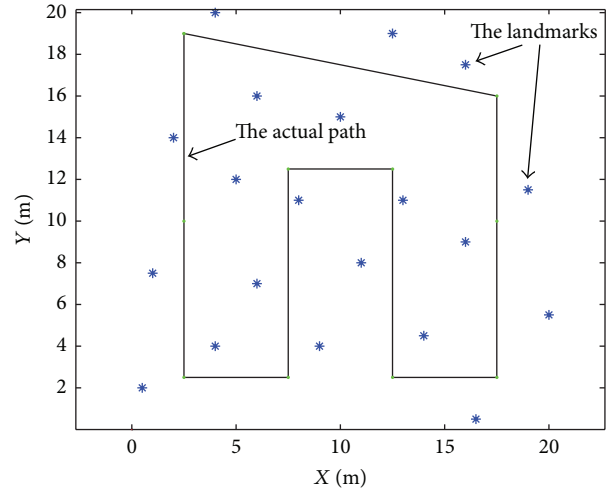


Figure 2: The simulation environment of the SLAM task.

Table 1: The values of parameters in the experiments.

Parameters	Value	Remarks
A_1 and A_2	1	The passive decay rate of the inputs to EKF
B_1 and B_2	1.5	The upper bounds of the inputs to EKF
D_1 and D_2	–1.5	The lower bounds of the inputs to EKF

For simplification without loss of generality, the noise is given out artificially in the simulation experiments. The specific type of sensors and the noise function for a given sensor are ignored in this paper. To show the advantages of the proposed bioinspired neural model based EKF approach (B-EKF), it is compared with the standard EKF algorithm (S-EKF) and the fuzzy logic based EKF algorithm (F-EKF). To remove the random effects on the SLAM algorithms, every experiment was conducted 15 times (about 5000 steps). The parameters in all of the experiments are the same, which are listed in Table 1.

3.1. The Experiment under Normal Noise Condition. In this experiment, the noise of the observation is under normal condition; namely, the noise will fluctuate in a little range during the whole moving process of the robot (see Figure 3). The results of the SLAM based on the proposed approach in this experiment are shown in Figure 4. Figure 5 is the localization error of the robot and the estimated error of the landmarks based on the S-EKF method, F-EKF method, and B-EKF method, respectively. The comparisons of the localization errors in the SLAM task are listed in Table 2.

Remark 3. The root mean squared error (RMSE) in Table 2 is calculated by

$$\mathrm{RMSE} = \sqrt{\frac{1}{n}\sum_{i=1}^{n}\left(e_i - \widehat{e}_i\right)^2}, \tag{14}$$

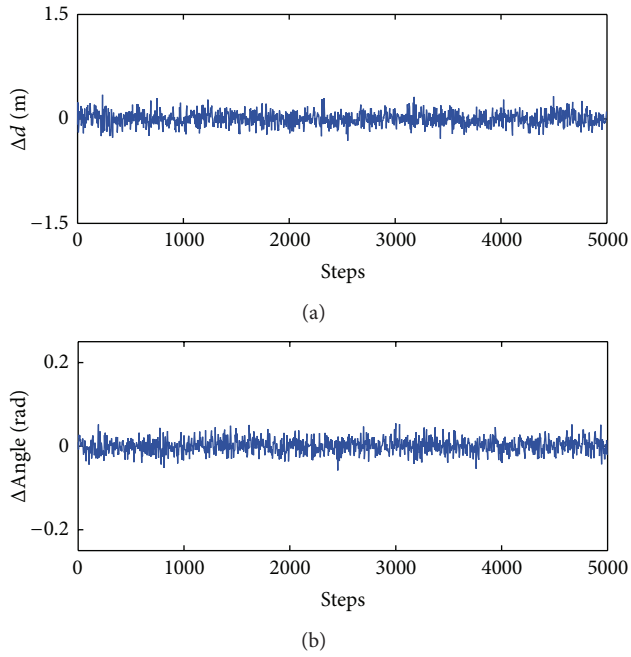


Figure 3: The noise distribution designed for the first experiment: (a) the error of the distance observation; (b) the error of the angle observation.

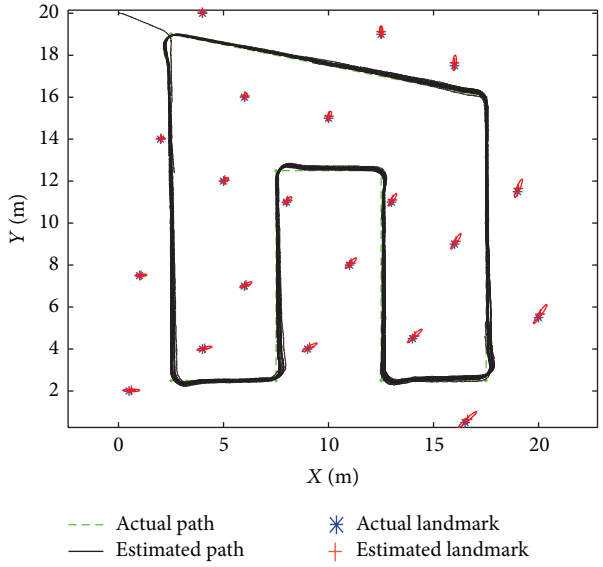


Figure 4: The results of the SLAM task based on the proposed approach in the first experiment.

where e_i and $\hat{e}_i$ are the real value and the predicted value of the robot position at the ith step, respectively.

The results in Figure 4 show that the trajectory calculated by B-EKF is closest to the actual trajectory of the robot, and the estimated landmarks are closest to the actual landmarks. The results of the localization error and the estimated error of the landmarks in Figure 5 show that all the three methods can complete the SLAM task under the normal noise condition.

Table 2: The root mean square error of the SLAM task in the first experiment.

The algorithms for the SLAM task	The root mean square error		
	ΔX/m	ΔY/m	ΔAngle/rad
The S-EKF based method	0.2707	0.2366	0.0829
The F-EKF based method	0.1205	0.1473	0.0472
The B-EKF based method	0.0976	0.0816	0.0105

Table 3: The root mean square error of the SLAM task in the second experiment.

The algorithms for the SLAM task	The root mean square error		
	ΔX/m	ΔY/m	ΔAngle/rad
The S-EKF based method	0.4064	0.3922	0.0921
The F-EKF based method	0.3083	0.3316	0.0496
The B-EKF based method	0.1132	0.0918	0.0204

However, the error of the proposed approach is less than both the F-EKF and S-EKF methods (see Table 2) and the fluctuation of errors in the proposed approach is very little, which shows that the proposed approach can deal with the SLAM problem stably (see Figure 5). The results in this experiment show that the proposed approach has a good performance to deal with the SLAM problem under normal noise condition.

3.2. The Experiment under Abnormal Noise Condition. To further test the performance of the proposed approach, this experiment is conducted, where the noise will fluctuate with the time violently (see Figure 6), which would happen if the performance of the onboard sensors becomes bad (such as the temperature effects). The results of the SLAM task based on the proposed approach are shown in Figure 7. Figure 8 is the localization error of the robot and the estimated error of the landmarks in this experiment. The comparisons of the localization errors in the SLAM task are listed in Table 3.

In this experiment, with the increase of noises, it is hard for the S-EKF method to adapt to this change; the reliability of the observations reduces and the filter becomes unstable, so the localization error and the estimated error will increase and fluctuate violently (see Figure 8(a)). In the F-EKF method, the error will increase obviously too (see Figure 8(b)), although it is less than the S-EKF method. The main reason is that the fuzzy rule in the F-EKF method is fixed, which cannot deal with this abnormal noise condition very well. However, in the proposed approach, with the increasing of noises, the bioinspired adaptive controller of the B-EKF method adjusts the weights of system noise and observation noise in real time, and the positioning accuracy will be significantly improved. During the whole exploration process of the robot, the localization error and the estimated error always fluctuate in a small range (see Figure 8(c)). The results in Figure 8 and Table 3 show that the proposed approach can deal with this abnormal noise condition in the SLAM task efficiently.

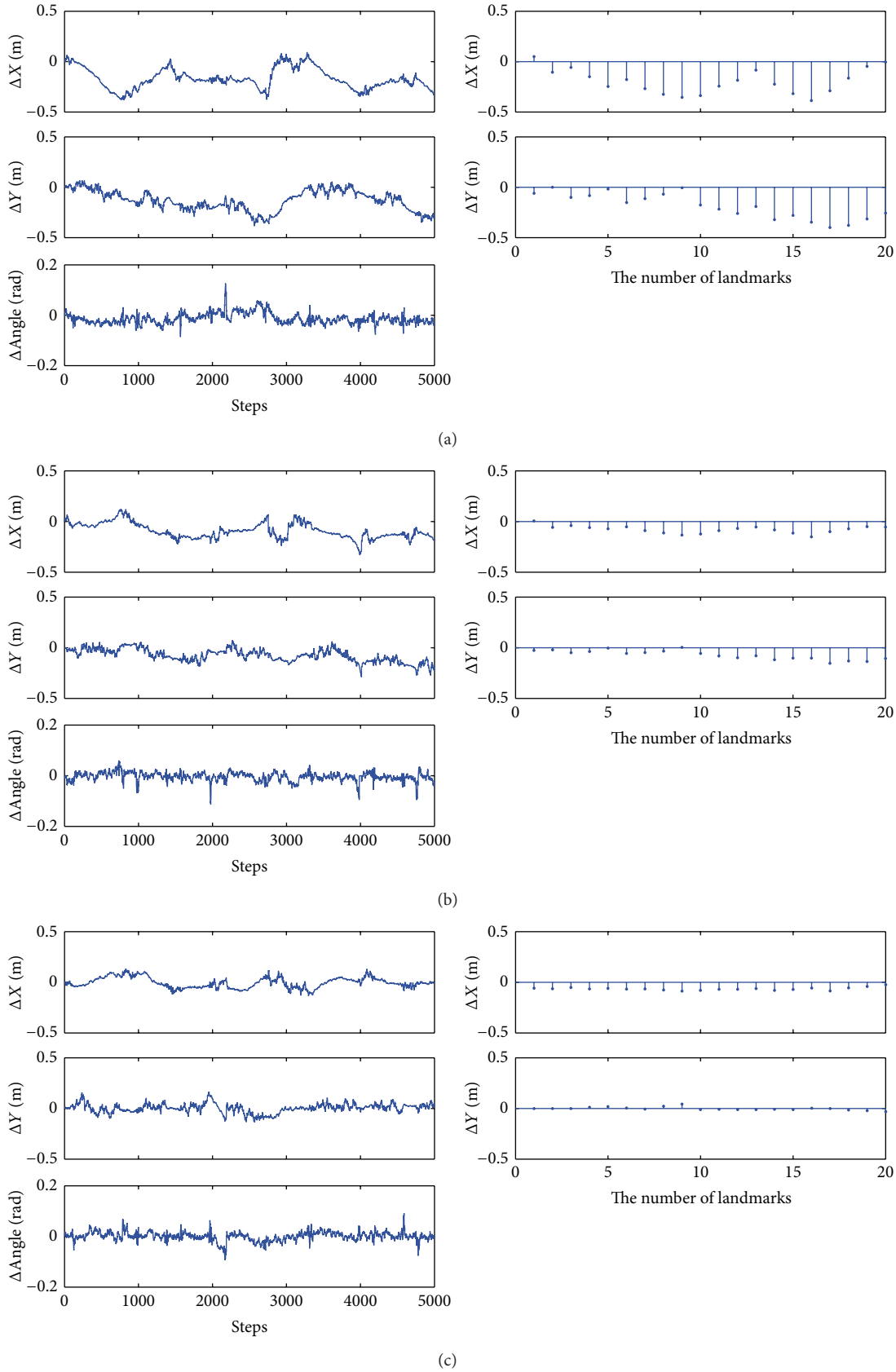


Figure 5: The localization error and the estimated error in the first experiment: (a) based on S-EKF method; (b) based on F-EKF method; (c) based on B-EKF method.

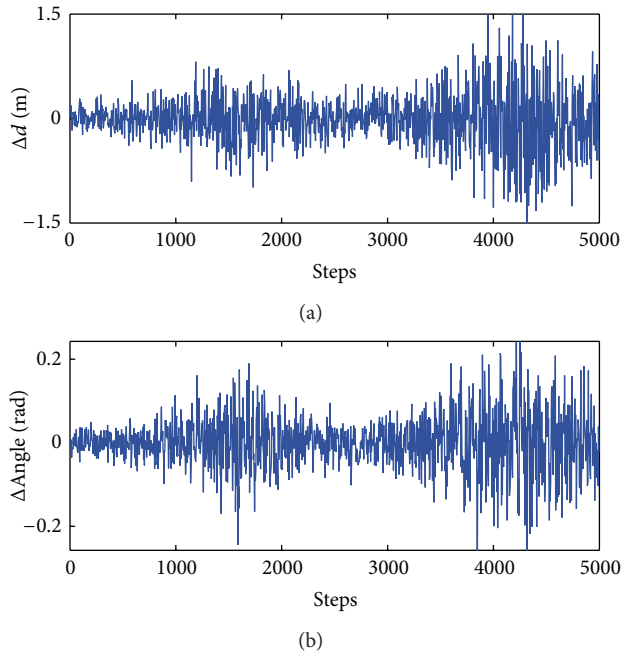


Figure 6: The noise distribution designed for the second experiment: (a) the error of the distance observation; (b) the error of the angle observation.

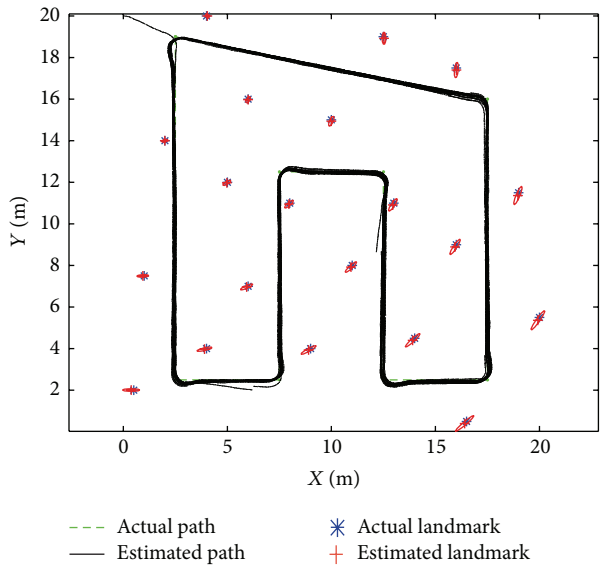


Figure 7: The results of the SLAM task based on the proposed approach in the second experiment.

4. Discussions

The results of the simulation experiments in Section 3 show that the proposed approach can satisfy the SLAM task under various situations. The parameter sensitivity and some performances of the proposed approach are discussed in this section.

Table 4: The root mean square error of the SLAM task by different parameters A_1 and A_2.

The value of the parameters A_1 and A_2	The root mean square error		
	ΔX/m	ΔY/m	ΔAngle/rad
0.5	0.1176	0.0982	0.0218
1	0.1132	0.0918	0.0204
1.5	0.1148	0.0942	0.0270
2	0.1262	0.1114	0.0385

At first, the parameters of the proposed approach are discussed. There are many discussions on the parameters of the bioinspired neural model in our previous work [41, 43, 45]. The upper and lower activity bounds B and D in the proposed bioinspired neural model will only effect the relative range of the weights $a(k)$ and $b(k)$ for the system noise and observation noise in the EKF based SLAM algorithm (see (8) and (12)), which can be decided by the real noise range in the SLAM tasks. So here just the parameters A_1 and A_2 are discussed, which are very important in the bioinspired neural model. To analyze the influence of parameters A_1 and A_2, some simulations were carried out under the same conditions as the second experiment in Section 3. The results are listed in Table 4. The results in Table 4 show that the proposed approach is not very sensitive to the variations of the parameters A_1 and A_2 even in the abnormal noise conditions. So the parameters can be chosen in a very wide range. All the cases studied in this paper use the same parameters, which are listed in Table 1.

To discuss the adaptability of the proposed approach in different SLAM tasks, two simulation experiments were conducted, where the noise distribution and the parameters of the proposed approach are the same as the second experiment in Section 3, except that the trajectory of the robot and the distribution of the landmarks are different. The results of these experiments based on the proposed approach are shown in Figure 9. The root mean square errors of the proposed approach in these SLAM tasks with the simple trajectory and the complex trajectory are {ΔX/m = 0.1020, ΔY/m = 0.0921, ΔAngle/rad = 0.0217} and {ΔX/m = 0.1103, ΔY/m = 0.0928, ΔAngle/rad = 0.0209}, respectively. The errors in the two experiments are both very little, which show that the proposed approach can deal with various SLAM tasks efficiently.

The process introduced in Section 2 and the simulation results of the SLAM tasks show that the proposed approach has some good performances. For example, the adaptivity of the proposed approach is good, which can adjust the filter in real time based on the noise conditions. Furthermore, the proposed approach can extend to real-world SLAM applications, and nothing needs to be done on the algorithm. The proposed approach is different from other algorithms, which need a priori knowledge or learning process [30, 46].

5. Conclusions

The EKF based SLAM approach of robot is investigated in this paper. When the noise model is unknown, the robustness

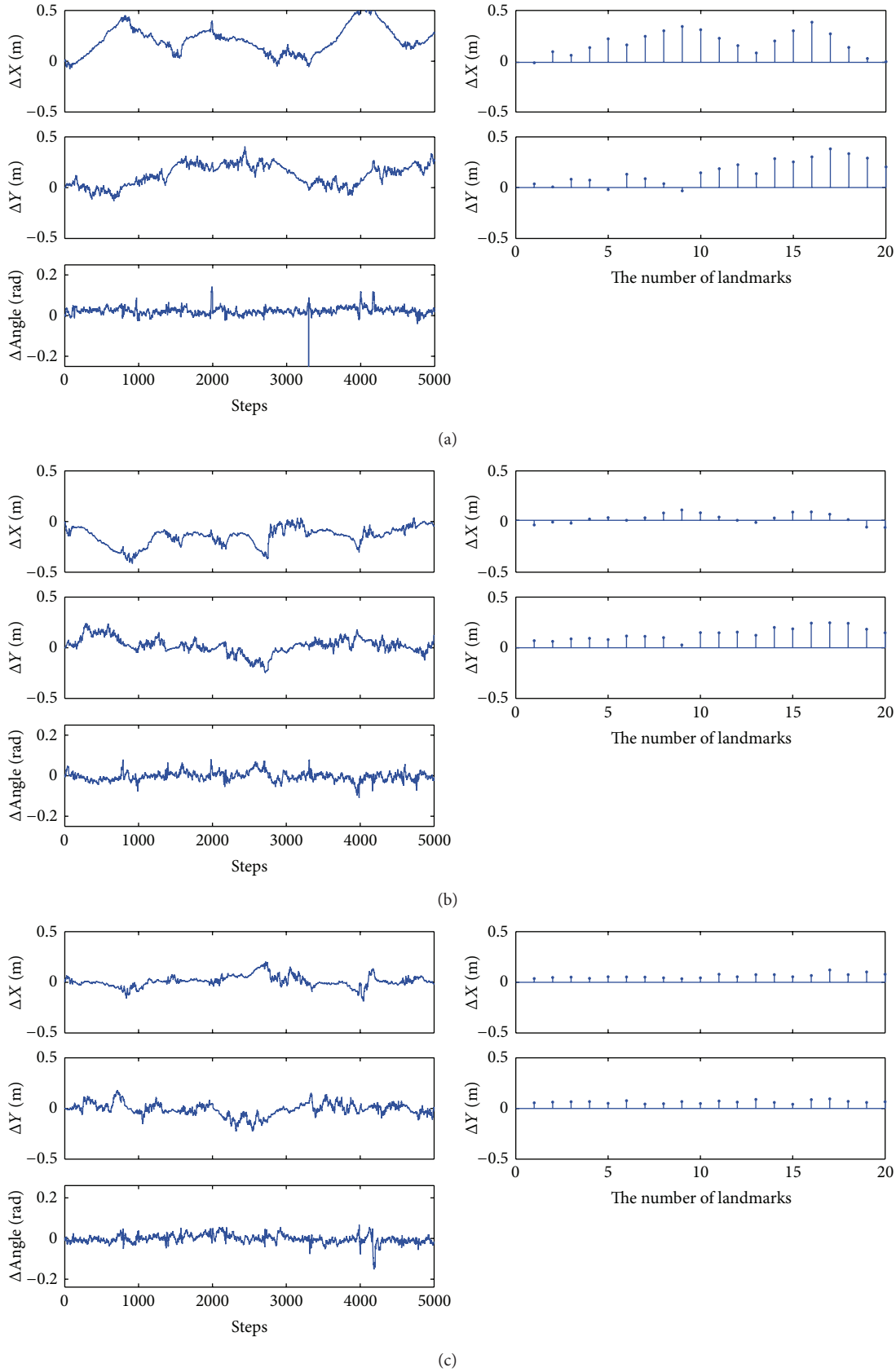


Figure 8: The localization error and the estimated error in the second experiment: (a) based on S-EKF method; (b) based on F-EKF method; (c) based on B-EKF method.

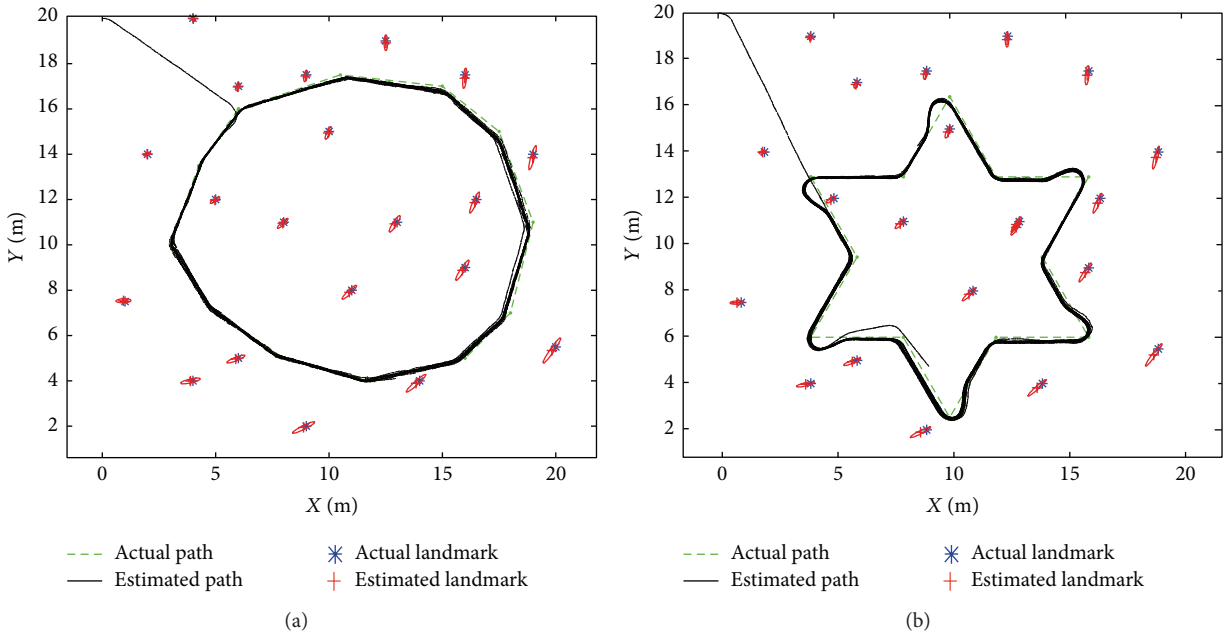


FIGURE 9: The results of the SLAM task based on the proposed approach with different trajectory and landmark distribution: (a) with a simple trajectory; (b) with a complex trajectory.

and accuracy of the EKF based SLAM algorithm will reduce. A novel adaptive EKF based SLAM algorithm structure is therefore proposed to adjust the weights of noises in the EKF based SLAM algorithm adaptively. Furthermore, a bioinspired neural model is integrated into this EKF based SLAM algorithm, which can reduce the error of the EKF based SLAM algorithm. As our approach is based on a bioinspired neural model, the SLAM task can be achieved efficiently, without any a priori knowledge of the noise model, nor any learning procedures. The feasibility and efficiency of the proposed approach have been discussed and illustrated through simulation studies. The proposed approach can deal with the searching and exploring problem in unknown environments, which has broad applications, such as the perceiving and understanding for the underwater and exoplanetary exploration.

In our research, there still remain some problems to be addressed. The first arises when the robot works in real and more complex environments. The second problem is the theoretical analysis of the bioinspired neural model based EKF for robot SLAM. In future work, some new bioinspired learning method and the 3D vision based method will be studied in the SLAM algorithm for robot.

Conflict of Interests

The authors declared that they have no conflict of interests to this work.

Acknowledgments

This work was supported by the National Natural Science Foundation of China (61203365), the Jiangsu Province Natural Science Foundation (BK2012149), the Open Fund of Changzhou Key Laboratory of Sensor Networks and Environmental Sensing (CZSN201102), the Fundamental Research Funds for the Central Universities (2011B04614), and the Science and Technology Commission of Shanghai Municipality (12595810200).

References

[1] M. W. M. Gamini Dissanayake, P. Newman, S. Clark, H. F. Durrant-Whyte, and M. Csorba, "A solution to the simultaneous localization and map building (SLAM) problem," *IEEE Transactions on Robotics and Automation*, vol. 17, no. 3, pp. 229–241, 2001.

[2] K. Çelik and A. K. Somani, "Monocular vision SLAM for indoor aerial vehicles," *Journal of Electrical and Computer Engineering*, vol. 2013, Article ID 374165, 15 pages, 2013.

[3] D. Ribas, P. Ridao, J. D. Tardós, and J. Neira, "Underwater SLAM in man-made structured environments," *Journal of Field Robotics*, vol. 25, no. 11-12, pp. 898–921, 2008.

[4] Z. Li, Z. Xie, and A. Ming, "Simultaneously firing sonar ring based high-speed navigation for non- holonomic mobile robots in unstructured environment," *International Journal of Vehicle Autonomous Systems*, vol. 6, no. 1, pp. 172–185, 2008.

[5] S. Lee, S. Lee, and S. Baek, "Vision-based kidnap recovery with SLAM for home cleaning robots," *Journal of Intelligent and Robotic Systems: Theory and Applications*, vol. 67, no. 1, pp. 7–24, 2012.

[6] J. Mullane, B. Vo, M. D. Adams, and B. Vo, "A random-finite-set approach to Bayesian SLAM," *IEEE Transactions on Robotics*, vol. 27, no. 2, pp. 268–282, 2011.

[7] W. L. D. Lui and R. Jarvis, "A pure vision-based topological SLAM system," *International Journal of Robotics Research*, vol. 31, no. 4, pp. 403–428, 2012.

[8] A. Chatterjee and F. Matsuno, "A neuro-fuzzy assisted extended kalman filter-based approach for simultaneous localization and mapping (SLAM) problems," *IEEE Transactions on Fuzzy Systems*, vol. 15, no. 5, pp. 984–997, 2007.

[9] W. Zhou, C. Zhao, and H. Zhang, "An AMPF and FastSLAM based compositive SLAM algorithm," *Pattern Recognition and Artificial Intelligence*, vol. 22, no. 5, pp. 718–725, 2009.

[10] M. Kaess, A. Ranganathan, and F. Dellaert, "iSAM: incremental smoothing and mapping," *IEEE Transactions on Robotics*, vol. 24, no. 6, pp. 1365–1378, 2008.

[11] D. Benedettelli, A. Garulli, and A. Giannitrapani, "Cooperative SLAM using M-Space representation of linear features," *Robotics and Autonomous Systems*, vol. 60, no. 10, pp. 1267–1278, 2012.

[12] J. Park, S. Lee, and J. Park, "Correction robot pose for SLAM based on extended Kalman filter in a rough surface environment," *International Journal of Advanced Robotic Systems*, vol. 6, no. 2, pp. 67–72, 2009.

[13] J. E. Guivant and E. M. Nebot, "Solving computational and memory requirements of feature-based simultaneous localization and mapping algorithms," *IEEE Transactions on Robotics and Automation*, vol. 19, no. 4, pp. 749–755, 2003.

[14] A. Gil, Ó. Reinoso, M. Ballesta, and M. Juliá, "Multi-robot visual SLAM using a Rao-Blackwellized particle filter," *Robotics and Autonomous Systems*, vol. 58, no. 1, pp. 68–80, 2010.

[15] S. Barkby, S. B. Williams, O. Pizarro, and M. V. Jakuba, "Bathymetric particle filter SLAM using trajectory maps," *International Journal of Robotics Research*, vol. 31, no. 12, pp. 1409–1430, 2012.

[16] J.-L. Blanco, J. González, and J.-A. Fernández-Madrigal, "Optimal filtering for non-parametric observation models: applications to localization and SLAM," *International Journal of Robotics Research*, vol. 29, no. 14, pp. 1726–1742, 2010.

[17] F. A. A. Cheein, F. M. L. Pereira, F. Di Sciascio, and R. Carelli, "Autonomous simultaneous localization and mapping driven by Monte Carlo uncertainty maps-based navigation," *Knowledge Engineering Review*, vol. 28, no. 1, pp. 35–57, 2013.

[18] M. Milford and G. Wyeth, "Persistent navigation and mapping using a biologically inspired SLAM system," *International Journal of Robotics Research*, vol. 29, no. 9, pp. 1131–1153, 2010.

[19] G. Wyeth and M. Milford, "Spatial cognition for robots: robot navigation from biological inspiration," *IEEE Robotics and Automation Magazine*, vol. 16, no. 3, pp. 24–32, 2009.

[20] A. Weitzenfeld, J.-M. Fellous, A. Barrera, and G. Tejera, "Allothetic and idiothetic sensor fusion in rat-inspired robot localization," in *Multisensor, Multisource Information Fusion: Architectures, Algorithms, and Applications*, vol. 8407 of *Proceedings of SPIE*, 2012.

[21] J. E. Guivant and E. M. Nebot, "Optimization of the simultaneous localization and map-building algorithm for real-time implementation," *IEEE Transactions on Robotics and Automation*, vol. 17, no. 3, pp. 242–257, 2001.

[22] H. Temeltas and D. Kayak, "SLAM for robot navigation," *IEEE Aerospace and Electronic Systems Magazine*, vol. 23, no. 12, pp. 16–19, 2008.

[23] L. Pedraza, D. Rodriguez-Losada, F. Matía, G. Dissanayake, and J. V. Miró, "Extending the limits of feature-based SLAM with B-splines," *IEEE Transactions on Robotics*, vol. 25, no. 2, pp. 353–366, 2009.

[24] L. de Souza Rosa and V. Bonato, "A method to convert floating to fixed-point EKF-SLAM for embedded robotics," *Journal of the Brazilian Computer Society*, vol. 19, no. 2, pp. 181–192, 2013.

[25] G. Sibley, C. Mei, I. Reid, and P. Newman, "Vast-scale outdoor navigation using adaptive relative bundle adjustment," *The International Journal of Robotics Research*, vol. 29, no. 8, pp. 958–980, 2010.

[26] P. Loncomilla and J. R. Del Solar, "Visual SLAM based on rigid-body 3D landmarks," *Journal of Intelligent and Robotic Systems: Theory and Applications*, vol. 66, no. 1-2, pp. 125–149, 2012.

[27] S. Lei and Z. Li, "Fusing visual tracking and navigation for autonomous control of an intelligen t wheelchair," in *Proceedings of the 3rd IFAC International Conference on Intelligent Control and Automation Science*, pp. 549–554, Chengdu, China, 2013.

[28] S. Tully, G. Kantor, and H. Choset, "A unified Bayesian framework for global localization and SLAM in hybrid metric/topological maps," *The International Journal of Robotics Research*, vol. 31, no. 3, pp. 271–288, 2012.

[29] F.-M. Eduardo, G.-J. Javier, and A. Vicente, "Creating metric-topological maps for large-scale monocular SLAM," in *Proceedings of the 10th International Conference on Informatics in Control, Automation and Robotics (ICINCO '13)*, vol. 2, pp. 39–47, Reykjavik, Iceland, 2013.

[30] H. Du, Y. Hao, and Y. Zhao, "SLAM algorithm based on fuzzy adaptive Kalman filter," *Journal of Huazhong University of Science and Technology (Natural Science Edition)*, vol. 40, no. 1, pp. 58–62, 2012.

[31] J. Kang, S. An, and S. Oh, "Modified neural network aided EKF based SLAM for improving an accuracy of the feature map," in *Proceeding of the International Joint Conference on Neural Networks (IJCNN '10)*, pp. 1–7, Barcelona, Spain, July 2010.

[32] M. J. Milford, *Robot Navigation from Nature: Simultaneous Localisation, Mapping, and Path Planning Based on Hippocampal Models*, Springer, 2008.

[33] M. Pfingsthorn and A. Birk, "Simultaneous localization and mapping with multimodal probability distributions," *International Journal of Robotics Research*, vol. 32, no. 2, pp. 143–171, 2013.

[34] H. Choi, D. Y. Kim, J. P. Hwang, C.-W. Park, and E. Kim, "Efficient simultaneous localization and mapping based on ceiling-view: ceiling boundary feature map approach," *Advanced Robotics*, vol. 26, no. 5-6, pp. 653–671, 2012.

[35] Y. Li, S. Li, and Y. Ge, "A biologically inspired solution to simultaneous localization and consistent mapping in dynamic environments," *Neurocomputing*, vol. 104, pp. 170–179, 2013.

[36] S. Huang and G. Dissanayake, "Convergence and consistency analysis for extended Kalman filter based SLAM," *IEEE Transactions on Robotics*, vol. 23, no. 5, pp. 1036–1049, 2007.

[37] H.-P. Li, D.-M. Xu, F.-B. Zhang, and Y. Yao, "Consistency analysis of EKF-based SLAM by measurement noise and observation times," *Acta Automatica Sinica*, vol. 35, no. 9, pp. 1177–1184, 2009.

[38] X. Ji, Z. Zheng, and H. Zhang, "Analysis and control of robot position error in SLAM," *Acta Automatica Sinica*, vol. 34, no. 3, pp. 323–330, 2008.

[39] S. X. Yang and M. Q.-. Meng, "Real-time collision-free motion planning of a mobile robot using a neural dynamics-based approach," *IEEE Transactions on Neural Networks*, vol. 14, no. 6, pp. 1541–1552, 2003.

[40] A. L. Hodgkin and A. F. Huxley, "A quantitative description of membrane current and its application to conduction and excitation in nerve," *The Journal of Physiology*, vol. 117, no. 4, pp. 500–544, 1952.

[41] J. Ni and S. X. Yang, "Bioinspired neural network for real-time cooperative hunting by multirobots in unknown environments," *IEEE Transactions on Neural Networks*, vol. 22, no. 12, pp. 2062–2077, 2011.

[42] H. Qu, S. X. Yang, A. R. Willms, and Z. Yi, "Real-time robot path planning based on a modified pulse-coupled neural network model," *IEEE Transactions on Neural Networks*, vol. 20, no. 11, pp. 1724–1739, 2009.

[43] C. Luo and S. X. Yang, "A bioinspired neural network for real-time concurrent map building and complete coverage robot navigation in unknown environments," *IEEE Transactions on Neural Networks*, vol. 19, no. 7, pp. 1279–1298, 2008.

[44] D. Zhu, Y. Zhao, and M. Yan, "A bio-inspired neurodynamics based back stepping path-following control of an AUV with ocean current," *International Journal of Robotics and Automation*, vol. 27, no. 3, pp. 298–307, 2012.

[45] X. Yang, *Neural network approaches to real-time motion planning and control of robotic systems [Ph.D. dissertation]*, Department of Electrical and Computer Engineering, University of Alberta, 1999.

[46] J. Kang, S. Kim, S. An, and S. Oh, "A new approach to simultaneous localization and map building with implicit model learning using neuro evolutionary optimization," *Applied Intelligence*, vol. 36, no. 1, pp. 242–269, 2012.

Hindawi Publishing Corporation
Mathematical Problems in Engineering
Volume 2014, Article ID 657985, 9 pages
http://dx.doi.org/10.1155/2014/657985

Research Article

A RBFNN-Based Adaptive Disturbance Compensation Approach Applied to Magnetic Suspension Inertially Stabilized Platform

Quanqi Mu,[1,2] Gang Liu,[1,2] and Xusheng Lei[1,2]

[1] *Science and Technology on Inertial Laboratory, Beihang University, Beijing 100191, China*
[2] *School of Instrument Science and Opto-Eletronics Engineering, Beihang University, Beijing 100191, China*

Correspondence should be addressed to Quanqi Mu; muquanqi369@163.com

Received 5 March 2014; Revised 11 June 2014; Accepted 11 June 2014; Published 3 August 2014

Academic Editor: Yi Chen

Compared with traditional mechanical inertially stabilized platform (ISP), magnetic suspension ISP (MSISP) can absorb high frequency vibrations via a magnetic suspension bearing system with five degrees of freedom between azimuth and pitch gimbals. However, force acting between rotor and stator will introduce coupling torque to roll and pitch gimbals. Since the disturbance of magnetic bearings has strong nonlinearity, classic state feedback control algorithm cannot bring higher precision control for MSISP. In order to enhance the control accuracy for MSISP, a disturbance compensator based on radial basis function neural network (RBFNN) is developed to compensate for the disturbance. Using the Lyapunov theorem, the weighting matrix of RBFNN can be updated online. Therefore, the RBFNN can be constructed without priori training. At last, simulations and experiment results validate that the compensation method proposed in this paper can improve ISP accuracy significantly.

1. Introduction

As the middle mechanism between aircraft and remote sensing payload, inertially stabilized platform (ISP) supports the payload, isolates aircraft movements (angular motion and linear motion), and holds the line-of-sight (LOS) of the payload vertically steady relative to earth [1–5]. It plays an important role in the remote sensing image quality. Therefore, GSM3000, SSM-270, SSM-350 [6, 7], PAV30, PAV80, PAV100 [8, 9], and T-AS mount [10] are developed in recent years.

However, most of the ISPs are suspended by mechanical bearings. Mechanical ISP absorbs high frequency vibrations passively via four dampers, but there is still a small proportion of residual disturbance that is transmitted to the payload through mechanical bearings. Therefore, the imaging quality is deteriorated. Magnetic suspension inertially stabilized platform (MSISP), which has the characteristic of frictionless and active vibration suppression capacity, can further enhance imaging quality. However, force acting between rotor and stator will introduce coupling torque to roll and pitch gimbals, increasing difficulty in gimbal control of ISP.

Currently, ISP control methods can be divided into three categories: (1) PID-based or intelligent PID control [11–13]. They are widely applied in practical engineering, but generally the control accuracy and antidisturbance capacity are not satisfactory. (2) Control methods based on disturbance compensation, attenuation, and rejection; feed-forward control, disturbance observer based control, and H_∞-based control, the typical literatures are listed as follows. Literature [3, 4] presented an acceleration feed-forward method to compensate for the disturbing torque caused by the linear acceleration of the carrier. But the signal of the accelerometer may be easily distorted by centripetal acceleration. In literature [14], a reduced-order disturbance observer (DOB) for line-of-sight control system was proposed. It presented a straightforward generic design approach and considered system parameter sensitivity and engineering implementation. Although the simulations validated system performance and robustness, the noise coupling would be potentially introduced by the observer. Literature [15] introduced an H_∞/predictive output control method for a three-axis gyrostabilized platform, and in the severe disturbance environment the proposed controller worked better than nonlinear proportion integration

differentiation (NLPID) controller. Unfortunately, complex design procedure limited its practical applications. Compared with the PID-based control, the control methods based on disturbance compensation, attenuation, and rejection could enhance the accuracy to some extent, but they are dependent on system model. (3) Advanced intelligent control: fuzzy control [16, 17] sliding mode control [18]. In recent years, neural-network-based control has been widely used in induction motor control [19, 20] and multiaxis motion control [21]. It can get much more attractive performance and robustness than conventional control method. However, it needs a lot of time and data to train the weighting of the neural network.

In order to enhance the accuracy and the environmental adaptability, a new RBFNN-based adaptive control approach for MSISP is proposed. It can estimate and eliminate disturbance effectively to improve control performance. At last, simulations and experiment results prove that the RBFNN-based adaptive control approach proposed can lead to precise control performance for the MSISP system.

2. Working Principle of the MSISP

Figure 1 shows the structure of three-axis MSISP. There are three gimbals: roll gimbal (outer gimbal), pitch gimbal (middle gimbal), and azimuth gimbal (inner gimbal). Both the roll gimbal and the pitch gimbal use indirect drive method (motor plus a pair of gears), and bearings of the two gimbals are mechanical bearings. As for the azimuth gimbal, it uses direct drive with a brushless DC permanent-magnetic synchronous motor, and its bearing is a maglev bearing. Gimbal servo controller generates control signals to drive the motors according to position and attitude provided by POS (position and orientation system), as well as the angular rate measured by rate gyro G_{xy} and G_z. Torque produced by motors compensates for outer disturbances and rotates the gimbals, thus making payload LOS tracking command position and attitude. Maglev bearing controller generates coil current control signals to adjust the distance between rotor and stator for actively absorb the residual high frequency vibrations.

Because the bandwidth of the maglev bearing control system is far beyond that of the gimbal control system, besides, based on PID control method, the angular error between the maglev bearing stator and rotor can be controlled within 0.003° that can be ignored, the gimbal control system can be considered to be independent of the maglev bearing system, and the force between the maglev bearing stator and rotor is considered as disturbance to the gimbal system.

According to the typical photo flight situation, the dynamic rotation range of the base plate is within ±5°. Furthermore, the aircraft angular rate is less than 10°/s under typical aircraft photo flight frequency spectrum (≤0.5 Hz) [6, 8], so the coupling torque among gimbals and the comprehensive influence from the base plate to each gimbal can be ignored [4, 22]. Therefore, the controller for gimbal can be designed separately.

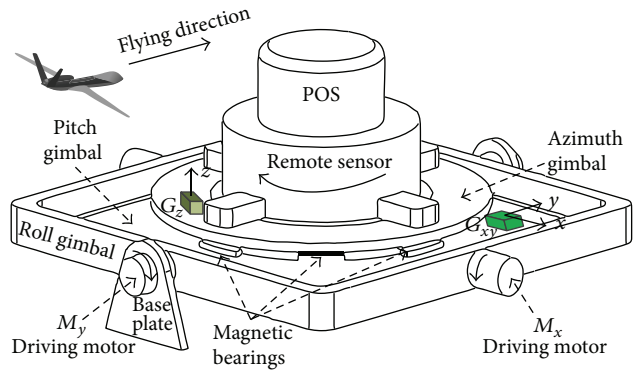


Figure 1: MSISP configuration diagram.

As for a single ISP gear-gimbal assembly driven by a DC torque motor with a pair of gears, its model can be described as follows:

$$\begin{aligned}\omega_L(s) &= \left(NK_tU(s) + \left(NT_{dm} + T_{dL}\right)\left(L_ms + R_m\right)\right. \\ &\quad \left. + \left[N(N-1)J_ms + N^2K_tK_e\right]\omega_b(s)\right) \\ &\quad \times \left(s\left(L_ms + R_m\right)\left(J_L + N^2J_m\right) + N^2K_tK_e\right)^{-1},\end{aligned} \tag{1}$$

where ω_L is the angular rate of the gimbal, K_t is the torque sensitivity, K_e is the back electromotive force (EMF) constant, R_m is the motor resistance, L_m is the motor inductance, U is the total voltage applied on the motor armature, T_M is the torque generated by motor, T_{dL} is the torque disturbances imposed on the payload, including imbalance, cable flexure, friction, and the force between the stator and the rotor of the maglev bearings, T_{dm} is the torque disturbances imposed on the motor, which are caused by bearing friction, cogging, and imperfections in the motor, N is the gear ratio, and ω_b is the angular rate of the base plate [2].

3. The RBFNN-Based Adaptive Controller Design

3.1. The State Space Modeling for Single ISP Gimbal. In actual system, the armature inductance L_m is so small that it usually can be ignored in the motor model. The angular rate is measured by gyro, while the attitude angular is measured by POS. In order to eliminate the steady error and to compensate for the attitude error caused by gyro drift error, select the state variables as

$$x = \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{bmatrix}, \tag{2}$$

where $x_1 = \omega_L$, $x_2 = \theta_L$, $\dot{x}_3 = x_2$, and $\dot{x}_4 = x_3$. State space equations of single gimbal are as follows:

$$\begin{aligned}
\dot{x} &= \begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \\ \dot{x}_3 \\ \dot{x}_4 \end{bmatrix} \\
&= \begin{bmatrix} \dfrac{-N^2K_tK_e}{JR_m}x_1 + \dfrac{NK_t}{JR_m}U + \dfrac{N^2K_tK_e}{JR_m}\omega_b + \dfrac{1}{J}T_d \\ x_1 \\ x_2 \\ x_3 \end{bmatrix} \\
&= \begin{pmatrix} \dfrac{-N^2K_tK_e}{JR_m} & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{pmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{bmatrix} \\
&\quad + \begin{pmatrix} \dfrac{NK_t}{JR_m} \\ 0 \\ 0 \\ 0 \end{pmatrix} U + \begin{pmatrix} \dfrac{N^2K_tK_e}{JR_m} & \dfrac{1}{J} \\ 0 & 0 \\ 0 & 0 \\ 0 & 0 \end{pmatrix} \begin{bmatrix} \omega_b \\ T_d \end{bmatrix} \\
&\triangleq Ax + Bu + d,
\end{aligned} \tag{3}$$

$$y = \theta_L = x_2 = \begin{pmatrix} 0 & 1 & 0 & 0 \end{pmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{bmatrix} \triangleq Cx,$$

where

$$\begin{aligned}
A &= \begin{pmatrix} \dfrac{-N^2K_tK_e}{JR_m} & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{pmatrix}, \\
B &= \begin{pmatrix} \dfrac{NK_t}{JR_m} \\ 0 \\ 0 \\ 0 \end{pmatrix}, \\
C &= \begin{pmatrix} 0 & 1 & 0 & 0 \end{pmatrix}, \\
d &= \begin{pmatrix} \dfrac{N^2K_tK_e}{JR_m} & \dfrac{1}{J} \\ 0 & 0 \\ 0 & 0 \\ 0 & 0 \end{pmatrix} \begin{bmatrix} \omega_b \\ T_d \end{bmatrix},
\end{aligned} \tag{4}$$

$$T_d = NT_{dm} + T_{dL} + N(N-1)J_m\dot{\omega}_b,$$

$$J = N^2J_m + J_L.$$

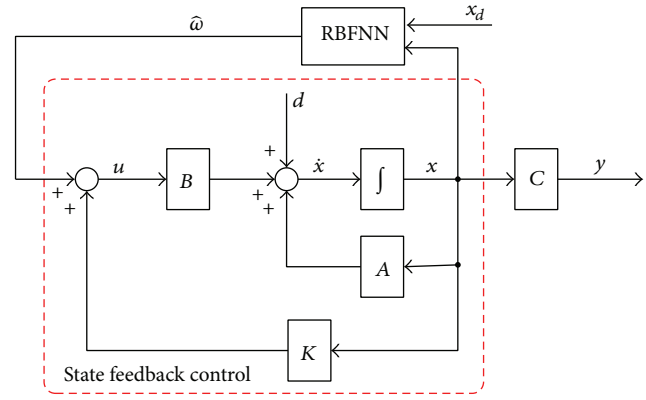


Figure 2: The RBFNN-based disturbance compensation scheme.

From model (1), the control input U is used to ensure LOS angular motion accuracy of the gimbal and to compensate the disturbance caused by ω_b, $\dot{\omega}_b$, T_{dm}, and T_{dL} at the same time. In order to enhance the accuracy of the MSISP, a RBFNN-based adaptive control approach is proposed. With the classic state feedback control, a disturbance compensator based on RBFNN is designed, which produces an additional equivalent control input based on the current system state and the desired state, thus realizing disturbance compensation (Figure 2).

3.2. The RBFNN-Based Adaptive Controller. The description above shows that the MSISP gimbal system is controllable, and then control input can be divided into two parts: one is the state feedback control input $Kx(t)$ and the other is the disturbance compensation input $\widehat{\omega}(t)$, that is,

$$u(t) = Kx(t) + \widehat{\omega}(t), \tag{5}$$

where $\widehat{\omega}(t)$ is disturbance estimation input generated by the adaptive RBFNN and K is weight matrix of state feedback control.

So, the dynamic model of the ISP gimbal system can be transformed as follows:

$$\dot{x}(t) = (A + BK)x(t) + B\widehat{\omega}(t) + d(t). \tag{6}$$

Then the error equation is as follows:

$$\dot{e}(t) = (A + BK)e(t) + B\widehat{\omega}(t) - f, \tag{7}$$

where the error between the desired state variable x_d and the real state variable x is denoted as $e = x - x_d$. The unknown function $f = \dot{x}_d - d(t) - (A + BK)x_d$ contains x_d and $d(t)$. The disturbance $d(t)$ cannot be measured directly. Since RBFNN has universal approximation capabilities, it can estimate the unknown function f successfully. Therefore, $-d(t)$ can be estimated by $\widehat{\omega}(t)$.

With RBFNN, the unknown function f can be estimated:

$$f = \overline{W}^T s(x) + \overline{\delta}(x), \tag{8}$$

where $\overline{W}^T$ is the ideal neural network weight matrix and $\overline{\delta}$ is the offset vector of neural network output layer.

Therefore, the error dynamic of the ISP gimbal system in (7) can be transformed as follows:

$$\dot{e}(t) = (A + BK)\, e(t) + B\widehat{\omega}(t) - \overline{W}^T s(x) - \overline{\delta}(x). \quad (9)$$

Using $\widehat{W}$ to approximate $\overline{W}$, the error between the actual value and the ideal RBFNN can be expressed as

$$\widehat{W}^T s(x) - \overline{W}^T s(x) = \widetilde{W}^T s(x). \quad (10)$$

Considering the following control law and RBFNN weighting adaptation law:

$$\dot{\widehat{W}}_{*i} = -\Gamma_i s_*(x)\, e^T P_{*i}, \quad (11)$$

$$\begin{aligned}\widehat{\omega}_i(t) = {} & \left(B^T B\right)^{-1} B_{i*}^T \widehat{W}_{*i}^T s_*(x) \\ & - \alpha\left(B^T B\right)^{-1} B_{i*}^T \operatorname{sign}\left(e^T P_{*i} B_{i*}\right),\end{aligned} \quad (12)$$

where $\widehat{W}_{*i} = \left[\widehat{W}_{1i}\ \ \widehat{W}_{2i}\ \cdots\ \widehat{W}_{li}\right]^T$, l is the number of hidden nodes of RBFNN, B_{i*} denotes the ith row of B, and $s_*(x)$ denotes a column vector $\left[s_1(x)\ \ s_2(x)\ \cdots\ s_l(x)\right]^T$. The disturbance can be approximated and eliminated by the adaptive RBFNN and the tracking error $e \to 0$, as $t \to \infty$.

Proof. See the appendix. □

4. Simulations and Experiments

4.1. Matlab Simulations. The effectiveness of the proposed RBFNN-based disturbance compensation scheme is evaluated by simulation results. Relevant parameters are listed in Table 1.

In order to verify the proposed scheme and compare different system performances, two different control methods are used in simulation.

(I) Conventional State Feedback Control. Based on the pole assignment theory, the matrices $A \in R^{4\times4}$ and $B \in R^{4\times1}$ and the state feedback matrix $K \in R^{1\times4}$ are selected as follows:

$$A = \begin{pmatrix} -1.2059 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{pmatrix},$$

$$B = \begin{pmatrix} 0.4422 \\ 0 \\ 0 \\ 0 \end{pmatrix},$$

$$K = \begin{pmatrix} -45.3092 & -4549 & -1813.3 & -90.6184 \end{pmatrix}. \quad (13)$$

(II) The RBFNN-Based Disturbance Compensation Scheme. Feedback controller parameters in this scheme are the same as the parameters of (I). In order to reduce calculations, only 10 hidden nodes of the adaptive RBFNN are selected, with $\alpha = 1.9375$.

Table 1: Parameters of a single gimbal system.

Parameter	Value	Unit
Motor torque constant	$K_t = 0.414$	Nm/Amp
Back EMF constant	$K_e = 0.407$	V/rad/s
Armature resistance	$R_m = 2.6$	Ohms
Motor inertial	$J_m = 2.765 \times 10^{-4}$	Kg m^2
Payload inertial	$J_L = 2.4$	Kg m^2
Drive ratio	$N = 6.7$	

According to the previous experience and actual flight tests, the disturbances can be separated as the sinusoidal eccentric disturbances and random friction disturbances [14]. (I) Sinusoidal part: the maximum payload of ISP is 100 kg, with maximum eccentric distance 5 mm and maximum disturbance translational acceleration ±1 g. As a result, the maximum imbalance torque is 5 Nm, and its frequency is selected as the maximum value of aircraft photo flight frequency (0.5 Hz). (II) Random friction part: the magnitude of the friction torque disturbance is considered as 1/4 of the sinusoid part. Then the external disturbing torque can be described as the following expression:

$$T_d = 5\sin(\pi t) + 2.5\,(\text{rand}(1) - 0.5), \quad (14)$$

where rand(1) stands for a random value within 0 and 1. Besides, set the base plate angular velocity as $\omega_b = 1.5 \times \pi/180 \times \sin(0.5 \times 2 \times \pi \times t)$ rad/s (amplitude is 1.5° and frequency is 0.5 Hz). Figure 3 shows the variation curve of disturbing torque.

The curves of attitude angles for two different control methods are shown in Figure 4. The attitude stabilization accuracy of MSISP system controlled by the RBFNN-based control scheme is 0.0169° (rms), which is 40% of the value generated by the state feedback control. The maximum attitude errors of the two control schemes are, respectively, 0.067° and 0.0298°. Additionally, the attitude curve with the proposed control scheme is more smoother than that of the state feedback scheme.

4.2. Experimental Setup. In order to validate the RBFNN-based disturbance compensation method in practical engineering applications, the sway table test is conducted. The MSISP is shown in Figure 5. The dimensions of the MSISP are 753 mm × 630 mm × 225 mm, and its weight is 91.5 kg. Figure 5(c) shows the detail configuration of the maglev bearings system. Different from the mechanical bearing suspension ISP, MSISP can achieve the target of noncontact suspension between the pitch gimbal ③ and the azimuth gimbal ⑥. ① is the radial bearing stator, and ② is the radial bearing rotor. They compose the radial magnetic bearings. ④ and ⑤ compose the upside axial magnetic bearings pair, while ⑦ and ⑧ compose the underside pair. ⑨ and ⑩, respectively, are the rotor and stator of the azimuth direct-driven motor.

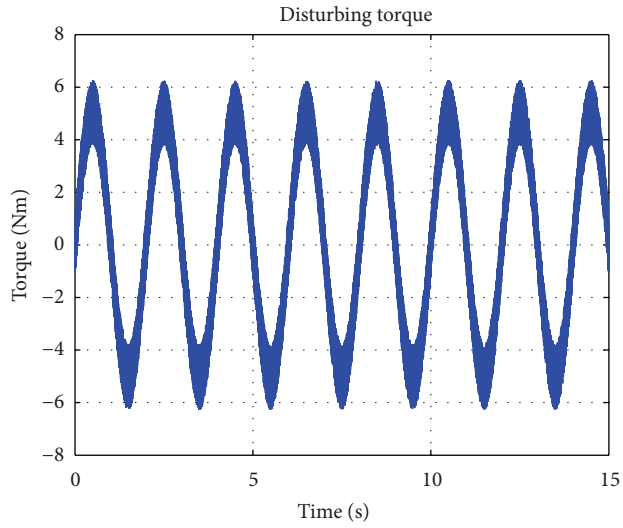


Figure 3: The variation curve of disturbing torque.

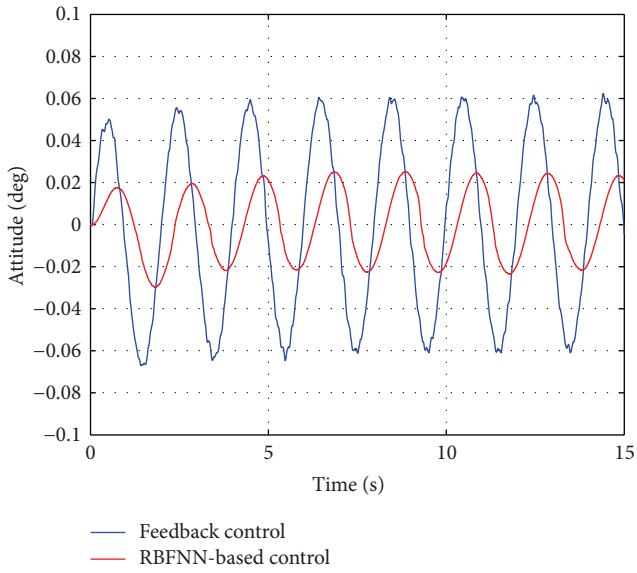


Figure 4: Attitude control curves of the two different control schemes.

Six degrees of freedom sway test table (E-Cue 624-1800), produced by Dutch FCS Corporation, can simulate attitude angular motion of the aircraft, and it is capable of handling a gross moving load (GML) of 1800 kg. Therefore, the simulation test environment based on six degrees of freedom is shown in Figure 6. The detail motion parameters are listed in Table 2. The MSISP is mounted on the top of the six degrees of freedom sway table. To validate the control performance, POS AV610, product of Applanix Corporation, is chosen as measurement, whose real-time attitude accuracy is 0.005° (RMS) [23]. Both ISP gimbal control system and maglev control system adopt floating-point DSP chip TMS320F28335. The servo control frequency reaches to 2 kHz. POS AV610 data is stored in the memory card for data processing after experiments. For safety reasons, each system adopts independent power supply. To simulate the airborne sensor, a series of simulated payloads are designed. The simulated payload on the ISP is 90 kg.

4.3. Experimental Results and Analysis. In the experiment, the sway table starts to work with small amplitude and then the amplitude is increased gradually until the expected sway extent. The sway curve of the test table is shown in Figure 7. Both control methods are programmed into ISP control programs, which are distinguished by control command. Different commands bring different control methods and the tested attitude curve for MSISP is shown in Figure 8. From Figures 7 and 8, the isolation role of the ISP can be shown. While the base plate sways largely as shown in Figure 7, the ISP can still hold the attitude of the sensor near 0° (Figure 8).

Figure 9 shows the comparison curves for two different control methods. Compared with the classic feedback control method, the RBFNN-based control method provides a high performance disturbance observer, so in theory, the performance of the latter scheme is higher than the former. In fact, the classic state feedback control has a certain compensation effect on disturbance, and the proposed RBFNN-based compensation method is used to further compensate the residual disturbance which is beyond the compensation capacity of the state feedback control. Compared with the classic state feedback control, the RBFNN-based compensation method can improve ISP attitude precision significantly. The precision and the maximum attitude angular generated by the RBFNN-based compensation method are 0.0111° (rms) and 0.046° that are nearly 43 percent and 53 percent of the values generated by the state feedback control method.

It is proved that the RBFNN-based control method is better than state feedback control on disturbance compensation. A suitable choosing for the node number and parameter α can achieve a much better effect.

5. Conclusions

To enhance attitude stabilization accuracy for three-axis MSISP, this paper proposes a novel RBFNN-based control scheme. Compared with the conventional state feedback control, the proposed scheme is effective in engineering application with easy implementation. Both simulations and flight tests validated that the proposed scheme could enhance the attitude stabilization accuracy significantly, and the steady accuracy of the MSISP control system is less than 0.02° (rms). Moreover, the proposed RBFNN-based control scheme can offer reference for other servo control systems such as turntable control, control moment gyro gimbal control, hard disk drive control, and so on.

Table 2: The parameters of the sway test table.

	Motion dynamics		Velocity	Acceleration
	Typical parameters	Absolute parameters		
Surge	−0.46~+0.57 m	−0.57~+0.57 m	±0.7 m/s	±7 m/s^2
Sway	±0.47 m	±0.50 m	±0.7 m/s	±7 m/s^2
Heave	±0.39 m	±0.39 m	±0.5 m/s	±10 m/s^2
Roll	±32.2°	±32.8°	±34°/s	>225°/s^2
Pitch	−23.2°~+25.5°	−27.4°~+31.6°	±34°/s	>225°/s^2
Yaw	±24.3°	±27.6°	±35°/s	>225°/s^2

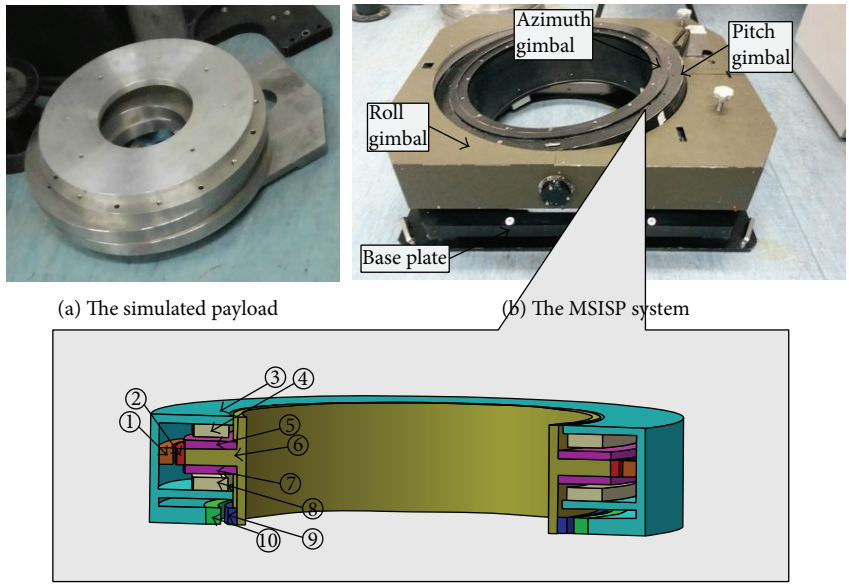


Figure 5: The simulated payload and MSISP system.

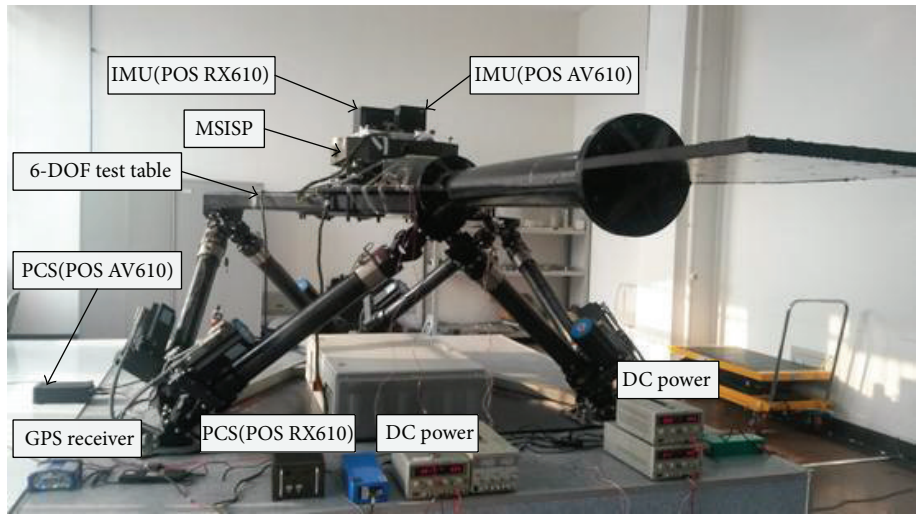


Figure 6: Sway test experiment system.

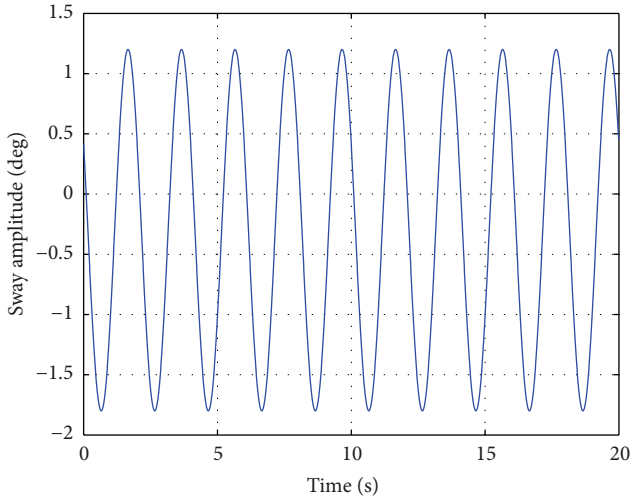


Figure 7: Test table attitude curve.

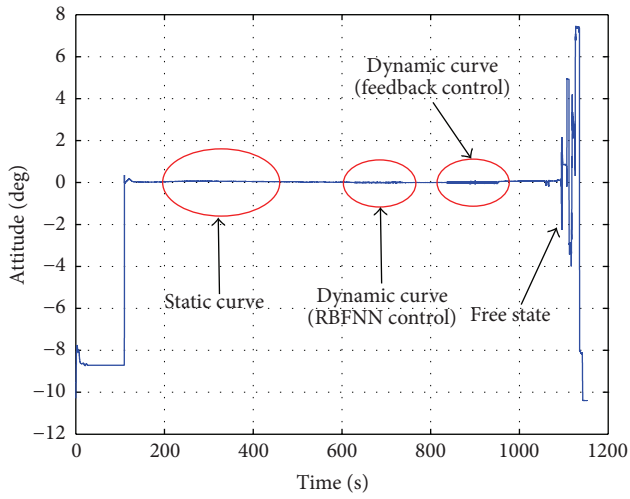


Figure 8: Sway experimental attitude wave.

Appendix

Proof of the asymptotically stable for disturbances estimation and elimination based on adaptive RBFNN.

Proof. Consider the following Lyapunov function:

$$V = e^T P e + \sum_{i=1}^{n} \widetilde{W}_{*i}^T \Gamma_i^{-1} \widetilde{W}_{*i}, \tag{A.1}$$

where $\widetilde{W}_{*i}^T = \left[\widetilde{W}_{1i} \;\; \widetilde{W}_{2i} \;\; \cdots \;\; \widetilde{W}_{li}\right]$, l is the number of hidden nodes of RBFNN, and Γ_i^{-1} is a dimensional compatible symmetric positive definite matrix. And the P is the symmetric definite matrix solution to

$$(A + BK)^T P + P(A + BK) = -Q, \tag{A.2}$$

where Q is the symmetric definite matrix solution.

Differing the function (A.1),

$$\begin{aligned} \dot{V} = & -e^T Q e + 2e^T P \left(B\widehat{\omega} - \overline{W}^T s(x) - \overline{\delta}(x)\right) \\ & + 2\sum_{i=1}^{n} \widetilde{W}_{*i}^T \Gamma_i^{-1} \dot{\widetilde{W}}_{*i}. \end{aligned} \tag{A.3}$$

Since $\overline{W}$ is a constant value, it is clear that

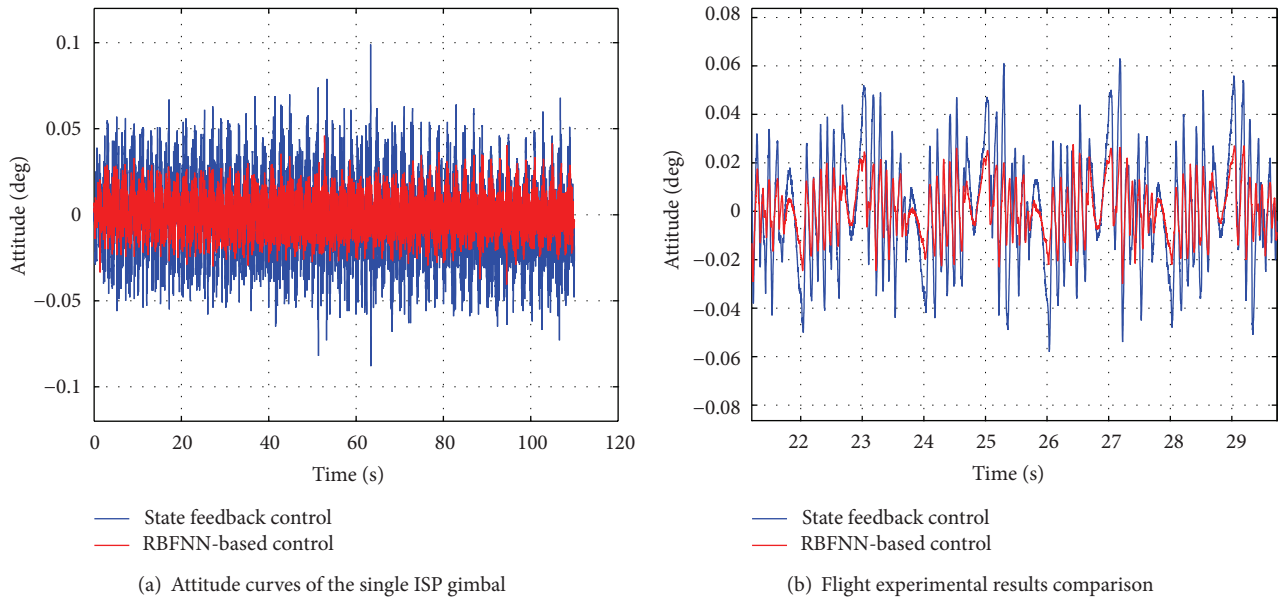
$$\dot{\widetilde{W}} = \dot{\widehat{W}}. \tag{A.4}$$

Since

$$\widehat{W}_{*i}^T s_*(x) - \widetilde{W}_{*i}^T s_*(x) = \overline{W}_{*i}^T s_*(x), \tag{A.5}$$

therefore, (A.3) is transformed as

$$\begin{aligned} \dot{V} &= -e^T Q e + 2e^T P \left(B\widehat{\omega} - \widehat{W}^T s(x) + \widetilde{W}^T s(x) - \overline{\delta}(x)\right) \\ &\quad + 2\sum_{i=1}^{n} \widetilde{W}_{*i}^T \Gamma_i^{-1} \dot{\widehat{W}}_{*i} \\ &= -e^T Q e + 2\left[e^T P_{*1} \;\; e^T P_{*2} \;\; \cdots \;\; e^T P_{*n}\right] \\ &\quad \times \begin{bmatrix} B_{1*}\widehat{\omega} - \widehat{W}_{*1}^T s_*(x) + \widetilde{W}_{*1}^T s_*(x) - \overline{\delta}_1(x) \\ B_{2*}\widehat{\omega} - \widehat{W}_{*2}^T s_*(x) + \widetilde{W}_{*2}^T s_*(x) - \overline{\delta}_2(x) \\ \vdots \\ B_{n*}\widehat{\omega} - \widehat{W}_{*n}^T s_*(x) + \widetilde{W}_{*n}^T s_*(x) - \overline{\delta}_n(x) \end{bmatrix} \\ &\quad + 2\sum_{i=1}^{n} \widetilde{W}_{*i}^T \Gamma_i^{-1} \dot{\widehat{W}}_{*i} \\ &= -e^T Q e + 2\sum_{i=1}^{n} e^T P_{*i} \\ &\quad \times \left(B_{i*}\widehat{\omega} - \widehat{W}_{*i}^T s_*(x) + \widetilde{W}_{*i}^T s_*(x) - \overline{\delta}_i(x)\right) \\ &\quad + 2\sum_{i=1}^{n} \widetilde{W}_{*i}^T \Gamma_i^{-1} \dot{\widehat{W}}_{*i} \\ &= -e^T Q e + 2\sum_{i=1}^{n} e^T P_{*i} \left(B_{i*}\widehat{\omega} - \widehat{W}_{*i}^T s_*(x) - \overline{\delta}_i(x)\right) \\ &\quad + 2\sum_{i=1}^{n} e^T P_{*i} \widetilde{W}_{*i}^T s_*(x) + 2\sum_{i=1}^{n} \widetilde{W}_{*i}^T \Gamma_i^{-1} \dot{\widehat{W}}_{*i} \\ &= -e^T Q e + 2\sum_{i=1}^{n} e^T P_{*i} \left(B_{i*}\widehat{\omega} - \widehat{W}_{*i}^T s_*(x) - \overline{\delta}_i(x)\right) \\ &\quad + 2\sum_{i=1}^{n} \left(\widetilde{W}_{*i}^T s_*(x) e^T P_{*i} + \widetilde{W}_{*i}^T \Gamma_i^{-1} \dot{\widehat{W}}_{*i}\right). \end{aligned} \tag{A.6}$$



(a) Attitude curves of the single ISP gimbal

(b) Flight experimental results comparison

Figure 9: The comparison of the sway experimental.

Submitting (11) and (12) into the (A.6) and choosing appropriate α to guarantee $\dot{V} \leq 0$

$$\begin{aligned}\dot{V} &= -e^T Q e + 2\sum_{i=1}^{n} e^T P_{*i}\left(B_{i*}\widehat{\omega} - \widehat{W}_{*i}^T s_* (x) - \overline{\delta}_i (x)\right) \\ &= -e^T Q e + 2\sum_{i=1}^{n} e^T P_{*i} B_{i*}\left(-\alpha \operatorname{sign}\left(e^T P_{*i} B_{i*}\right) - \overline{\delta}_i (x)\right) \\ &\leq -e^T Q e \leq 0.\end{aligned} \tag{A.7}$$

The error dynamic model of the ISP gimbal system is asymptotically stable.

Then, it can prove that the tracking error $e \to 0$, as $t \to \infty$.

Therefore, with the controller (5), (12), and the RBFNN weighting adaptation law (11), the system (6) is stable. The external disturbance can be approximated and eliminated by the adaptive RBFNN, and the tracking error $e \to 0$, as $t \to \infty$:

$$\lim_{t \to \infty} [x - x_d] = 0. \tag{A.8}$$

That is, asymptotic tracking is achieved. □

Conflict of Interests

The authors declare that there is no conflict of interests regarding the publication of this paper.

Acknowledgments

The project is supported in part by the National Basic Research Program of China (973 program; Grant no. 2009CB724001/2), in part by the National Natural Science Foundation of China under Grant 61273033, in part by the Beijing Municipal Natural Science Foundation under Grant 4132043, in part by the National Natural Science Foundation of China under Grant 61374029, and in part by the Beijing Municipal Natural Science Foundation under Grant 61374211.

References

[1] J. M. Hilkert, "Inertially stabilized platform technology: concepts and principles," *IEEE Control Systems Magazine*, vol. 28, no. 1, pp. 26–46, 2008.

[2] M. K. Masten, "Inertially stabilized platforms for optical imaging systems: tracking dynamic targets with mobile sensors," *IEEE Control Systems Magazine*, vol. 28, no. 1, pp. 47–64, 2008.

[3] M. Řezáč and Z. Hurák, "Vibration rejection for inertially stabilized double gimbal platform using acceleration feedforward," in *Proceedings of the 20th IEEE International Conference on Control Applications (CCA '11)*, pp. 363–368, Denver, Colo, USA, September 2011.

[4] Q. Mu, G. Liu, M. Zhong, and Z. Chu, "Imbalance torque compensation for three-axis inertially stabilized platform using acceleration feedforward," in *Proceedings of the 8th IEEE International Symposium on Instrumentation and Control Technology (ISICT '12)*, pp. 157–160, London, UK, July 2012.

[5] Y. Zhang, C. Du, and Q. Mu, "Random error modelling and compensation of accelerometer in airborne remote sensing stabilized platform," *Transactions of the Institute of Measurement and Control*, vol. 35, no. 4, pp. 503–509, 2013.

[6] http://www.npk-photonica.ru/images/specifications_gsm_3000.pdf.

[7] http://www.somag-ag.de/index.php?id=23&L=1.

[8] http://www.leica-geosystems.com/downloads123/zz/airborne/PAV80/Flyer/Leica_PAV80_Flyer_en.pdf.

[9] http://www.leica-geosystems.com/en/Leica-PAV100_103713.htm.

[10] http://www.geospace.co.za/pdf/DMC%20Brochure.pdf.

[11] W. Ji, Q. Li, and B. Xu, "Design study of adaptive fuzzy PID controller for LOS stabilized system," in *Proceedings of the 9th International Conference on Intelligent Systems Design and Applications (ISDA '06)*, pp. 336–341, October 2006.

[12] H. J. Hong, P. P. Yun, C. S. Zhao, and Q. Wu, "The application research on fuzzy PI control arithmetic of photoelectric stabilized platform," in *Proceedings of the International Workshop on Intelligent Systems and Applications (ISA '09)*, pp. 1–5, May 2009.

[13] A. R. Amr, S. Chengzhi, F. M. Hany, and W. Tongyu, "Design a robust PI controller for line of sight stabilization system," *International Journal of Modern Engineering Research (IJMER)*, pp. 144–148.

[14] J. M. Hilkert and B. Pautler, "A reduced-order disturbance observer applied to inertially stabilized Line-of-Sight control," in *Proceedings of the 25th SPIE Acquisition, Tracking, Pointing, and Laser Systems Technologies*, vol. 8052, 2011.

[15] R. D. Mahdy, A. N. Amir, and K. S. Ali, "Predictive output control of a three-axis gyrostabilized platform," *Proceedings of the IMechE, Part G: Journal of Aerospace Engineering*, pp. 1–11, 2013.

[16] K. C. Tan, T. H. Lee, E. F. Khor, and D. C. Ang, "Design and real-time implementation of a multivariable gyro-mirror line-of-sight stabilization platform," *Fuzzy Sets and Systems*, vol. 128, no. 1, pp. 81–93, 2002.

[17] J. A. R. Krishna Moorty, R. Marathe, and H. Babu, "Fuzzy controller for line-of-sight stabilization systems," *Optical Engineering*, vol. 43, no. 6, pp. 1394–1400, 2004.

[18] Y. B. Shtessel, "Decentralized sliding mode control in three-axis inertial platforms," *Journal of Guidance, Control, and Dynamics*, vol. 18, no. 4, pp. 773–781, 1995.

[19] T. Chen and T. Sheu, "Model reference neural network controller for induction motor speed control," *IEEE Transactions on Energy Conversion*, vol. 17, no. 2, pp. 157–163, 2002.

[20] M. Mohamadian, E. Nowicki, F. Ashrafzadeh, A. Chu, R. Sachdeva, and E. Evanik, "A novel neural network controller and its efficient DSP implementation for vector-controlled induction motor drives," *IEEE Transactions on Industry Applications*, vol. 39, no. 6, pp. 1622–1629, 2003.

[21] F. Lin and P. Shen, "Robust fuzzy neural network sliding-mode control for two-axis motion control system," *IEEE Transactions on Industrial Electronics*, vol. 53, no. 4, pp. 1209–1225, 2006.

[22] J. C. Fang, Z. H. Qi, and M. Y. Zhong, "Feedforward compensation method for three axes inertially stabilized platform imbalance torque," *Journal of Chinese Inertial Technology*, vol. 18, no. 1, pp. 38–43, 2010.

[23] http://www.optron.com/system-files/applanix-pos-av-dire-1304365131.pdf.

Hindawi Publishing Corporation
Mathematical Problems in Engineering
Volume 2014, Article ID 358269, 9 pages
http://dx.doi.org/10.1155/2014/358269

Research Article

Hybrid Taguchi DNA Swarm Intelligence for Optimal Inverse Kinematics Redundancy Resolution of Six-DOF Humanoid Robot Arms

Hsu-Chih Huang,[1] Sendren Sheng-Dong Xu,[2] and Huan-Shiuan Hsu[1]

[1] *Department of Electrical Engineering, National Ilan University, Yilan 26047, Taiwan*
[2] *Graduate Institute of Automation and Control, National Taiwan University of Science and Technology, Taipei 106, Taiwan*

Correspondence should be addressed to Hsu-Chih Huang; hchuang@niu.edu.tw

Received 20 February 2014; Revised 29 May 2014; Accepted 1 July 2014; Published 22 July 2014

Academic Editor: Yi Chen

This paper presents a hybrid Taguchi deoxyribonucleic acid (DNA) swarm intelligence for solving the inverse kinematics redundancy problem of six degree-of-freedom (DOF) humanoid robot arms. The inverse kinematics problem of the multi-DOF humanoid robot arm is redundant and has no general closed-form solutions or analytical solutions. The optimal joint configurations are obtained by minimizing the predefined performance index in DNA algorithm for real-world humanoid robotics application. The Taguchi method is employed to determine the DNA parameters to search for the joint solutions of the six-DOF robot arms more efficiently. This approach circumvents the disadvantage of time-consuming tuning procedure in conventional DNA computing. Simulation results are conducted to illustrate the effectiveness and merit of the proposed methods. This Taguchi-based DNA (TDNA) solver outperforms the conventional solvers, such as geometric solver, Jacobian-based solver, genetic algorithm (GA) solver and ant, colony optimization (ACO) solver.

1. Introduction

Recently, humanoid robotics has attracted much attention in the field of robotics. Humanoid robots are capable of performing various operations that are originally conceived for humans [1–5]. In contrast with the industrial robots, the humanoid robots emulate human's function to achieve a wide variety of tasks. These high-DOF humanoid robots are superior to conventional industrial robots in terms of flexibility, kinematics performance, agility, and dynamic performance. With these advantages, humanoid robotics has become a new challenging field in designing home service robotic systems [1–5].

Humanoid robot arm is one of the main studies of humanoid robotics. Generally, a humanoid robot arm is composed of six or seven DOFs from the shoulder to the wrist. From the kinematics point of view, the robot arm imitating the human arm motions is kinematically redundant like the human arm [6–8]. In other words, there is a kinematics redundancy because an infinite number of joint angles result in the same end-effector position. In kinematics analysis of humanoid robot arms, forward kinematics problems are straightforward and there is no complexity for deriving the kinematics equations [6–10]. However, inverse kinematics is a much more difficult problem compared with the forward kinematics [6–10].

The solution of the inverse kinematics problem is computationally intensive and generally takes a very long time in the real-time control of robot arms. The inverse kinematics mapping is in general one-to-many, involves complex inverse trigonometric functions, and has no closed-form solutions for the humanoid robot arms. This redundancy problem refers to the fact that the joint space of robot has a larger dimension than the task space [9, 10]. The inverse kinematics problem of the humanoid robots has been studied in the last decade [6–10]. Overall, the conventional analytical and Jacobian-based methods are generally computationally intensive and are not suitable for real-time humanoid robotics

applications. Moreover, these solvers did not provide a generalized optimal joint configuration of the high-DOF humanoid robot arm, meaning that these solvers are not applicable to other high-DOF robotic arms.

Swarm intelligence is a new category of methods in the optimization framework for solving the complex optimization problems in a wide variety of real-world applications [11–16]. Compared with the conventional methods, these approaches are more powerful, so that they do not need the reformulation of the problem to search a nonlinear space with real-world conditions [11–16]. Among these modern metaheuristic-based approaches for complex problems solving, DNA algorithm has been regarded as another efficient optimization technique. This algorithm proposed by Aldelman [17] emulates the concept of the bimolecular evolution and uses biomolecules for finding optimal solutions of complicated computational problems. This computing paradigm has successfully been used to solve complex problems in many disciplines [17–20] because they have more plentiful genetic information [17–20]. However, the parameters of the proposed DNA swarm intelligence algorithms are usually determined by trial-and-error approach [17–20]. These parameters influence the performance of the DNA algorithms. Unfortunately, the parameters are not appropriately set in the studies to solve complex optimization problems [17–20].

Taguchi method is a statistical method developed by Genichi Taguchi to improve the product quality and cost in industrial design [21–25]. This approach has proven to be an important tool in the system design and process quality in which the best setting of the control factors (parameters) is determined and has been applied to solve many optimization problems in electrical machines design, aerospace engineering, and controller design. For example, Su et al. [22] proposed an efficient GA approach combined with Taguchi method for mixed constrained circuit design. Yang et al. [23] employed the Taguchi method combined with GA to design a flight controller. Seenivasan et al. [24] presented an optimization of dehumidifier using Taguchi method. Hasanien [25] proposed an optimal controller in automatic voltage regulator system using Taguchi combined GA. Compared to the conventional full factorial design method, this method has fewer experiments to obtain optimal parameters or factors in optimization problems using the orthogonal array and signal-to-noise ratio techniques. To the authors' best understanding, there has been no attempt to developing hybrid Taguchi DNA swarm intelligences for the optimal inverse kinematics redundancy resolution of six-DOF humanoid robot arms.

The objective of this paper is to develop a hybrid Taguchi DNA algorithm to solve the inverse kinematics problem of the six-DOF humanoid robot arm more effectively. The rest of this paper is organized as follows. In Section 2, the forward kinematics and inverse kinematics of the six-DOF humanoid robot arm are introduced. Section 3 elucidates the procedure of how to apply the proposed Taguchi-based DNA swarm intelligence to solve the redundant problem and find an optimal configuration of the robot arm. Section 4 conducts several simulations to show the performance and merit of the proposed methods. Section 5 concludes this paper.

2. Forward Kinematics and Inverse Kinematics of the Six-DOF Humanoid Robot Arm

Forward kinematics and inverse kinematics are very important for studying the kinematics motion, path planning, and coordination of the humanoid robot arm. The joints information is necessary for these studies. Tasks to be performed by a robot arm are assigned in the Cartesian space which includes position vector $p \in \mathbf{R}^3$ (x, y, z), whereas robot actuators work in joint space is represented by joint angles $q \in \mathbf{R}^6$ $(\theta_1, \theta_2, \theta_3, \theta_4, \theta_5, \theta_6)$. The degree in joint space is three and the degree in Cartesian space is six; the inverse kinematics is, therefore, a redundancy problem. The forward kinematic equations relate joint motions of humanoid robot arm to the Cartesian coordinates. On the other hand, the conversion of the position of a humanoid robot arm end-effector from Cartesian space to joint space is called inverse kinematics problem [6–8].

A commonly used coordinate transformation for selecting frames of reference in robotic applications is the Denavit-Hartenberg (D-H) convention which involves the allocation of coordinate frames to each link. The positions of subsequent links are defined by the homogeneous transform matrix, which transforms the frame attached to link $i-1$ into a frame fixed to link i. In this transformation, each homogeneous transformation ${}^{i-1}A_i$ is represented as a product of four basic transformations, given by

$$
\begin{aligned}
{}^{i-1}A_i &= R_{z,\theta_i}\,\mathrm{Trans}_{z,d_i}\,\mathrm{Trans}_{x,a_i}R_{x,\alpha_i} \\
&= \begin{bmatrix} c\theta_i & -s\theta_i & 0 & 0 \\ s\theta_i & c\theta_i & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}
\begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & d_i \\ 0 & 0 & 0 & 1 \end{bmatrix} \\
&\quad \times \begin{bmatrix} 1 & 0 & 0 & a_i \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}
\begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & c\alpha_i & -s\alpha_i & 0 \\ 0 & s\alpha_i & c\alpha_i & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix},
\end{aligned} \tag{1}
$$

where the four quantities θ_i, a_i, d_i and α_i are parameters associated with link i and joint i. The four parameters θ_i, a_i, d_i, and α_i in (1) generally denote the joint angle, link length, link offset, and link twist of link i, respectively. $c\theta_i$, and $s\theta_i$ $(i = 1, 2, 3, \ldots, 6)$ denote $\cos\theta_i$ and $\sin\theta_i$. These parameters are derived from the geometric relationship between two coordinate frames [6–8]. Figure 1 depicts the picture of the general six-DOF humanoid robot arm. Table 1 lists all the D-H parameters of the six-DOF humanoid robot arm in this study. This mechanical information of the robot arm is needed to derive the forward kinematics of the robot arm.

Table 1: D-H parameters of the six-DOF humanoid robot arm.

i	θ_i	d_i	a_i	α_i	$\theta_{i,\min} < \theta_i < \theta_{i,\max}$
1	90°	0	L_0 (mm) 142.3	−90°	$-130° < \theta_1 < 130°$
2	90°	L_1 (mm) 145	0	−90°	$-80° < \theta_2 < 80°$
3	−90°	L_2 (mm) 120	0	−90°	$-130° < \theta_3 < 130°$
4	0°	L_3 (mm) 141	0	90°	$-170° < \theta_4 < -60°$
5	0°	L_4 (mm) 92	0	−90°	$-130° < \theta_5 < 130°$
6	90°	L_5 (mm) 172	0	−90°	$-140° < \theta_6 < -70°$

The transformation matrix between each two successive joint can be written by simply substituting the parameters from Table 1 into the matrix given as follows:

$$
{}^{0}A_1 = \begin{bmatrix} \cos(\theta_1) & 0 & -\sin(\theta_1) & L_0\cos(\theta_1) \\ \sin(\theta_1) & 0 & \cos(\theta_1) & L_0\sin(\theta_1) \\ 0 & -1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix},
$$

$$
{}^{1}A_2 = \begin{bmatrix} \cos(\theta_2) & 0 & -\sin(\theta_2) & 0 \\ \sin(\theta_2) & 0 & \cos(\theta_2) & 0 \\ 0 & -1 & 0 & L_1 \\ 0 & 0 & 0 & 1 \end{bmatrix},
$$

$$
{}^{2}A_3 = \begin{bmatrix} \cos(\theta_3) & 0 & -\sin(\theta_3) & 0 \\ \sin(\theta_3) & 0 & \cos(\theta_3) & 0 \\ 0 & -1 & 0 & L_2 \\ 0 & 0 & 0 & 1 \end{bmatrix},
$$

$$
{}^{3}A_4 = \begin{bmatrix} \cos(\theta_4) & 0 & \sin(\theta_4) & 0 \\ \sin(\theta_4) & 0 & -\cos(\theta_4) & 0 \\ 0 & 1 & 0 & L_3 \\ 0 & 0 & 0 & 1 \end{bmatrix},
$$

$$
{}^{4}A_5 = \begin{bmatrix} \cos(\theta_5) & 0 & -\sin(\theta_5) & 0 \\ \sin(\theta_5) & 0 & \cos(\theta_5) & 0 \\ 0 & -1 & 0 & L_4 \\ 0 & 0 & 0 & 1 \end{bmatrix},
$$

$$
{}^{5}A_6 = \begin{bmatrix} \cos(\theta_6) & 0 & -\sin(\theta_6) & 0 \\ \sin(\theta_6) & 0 & \cos(\theta_6) & 0 \\ 0 & -1 & 0 & L_5 \\ 0 & 0 & 0 & 1 \end{bmatrix}. \tag{2}
$$

Therefore, the transformation matrix of the end-effector with respect to the fixed reference frame can be expressed as follows:

$$
{}^{0}T_{\text{end}} = {}^{0}T_6 = {}^{0}A_1 \cdot {}^{1}A_2 \cdot {}^{2}A_3 \cdot {}^{3}A_4 \cdot {}^{4}A_5 \cdot {}^{5}A_6 = \begin{bmatrix} n_x & o_x & a_x & p_x \\ n_y & o_y & a_y & p_y \\ n_z & o_z & a_z & p_z \\ 0 & 0 & 0 & 1 \end{bmatrix}, \tag{3}
$$

where $p = [p_x, p_y, p_z]^T$ is the position of the end-effector in Cartesian coordinates and $o = \begin{bmatrix} n_x & o_x & a_x \\ n_y & o_y & a_y \\ n_z & o_z & a_z \end{bmatrix}$ is the rotation matrix. Although the orientation of the end-effector is generally of interest in a robotic application, most robot arms are used to perform simple and repetitive tasks. In these applications, the inverse kinematic solvers only consider the end-effector position matrix; that is, the Euler angles are not included and the rotation matrix of end-effector is not considered [26].

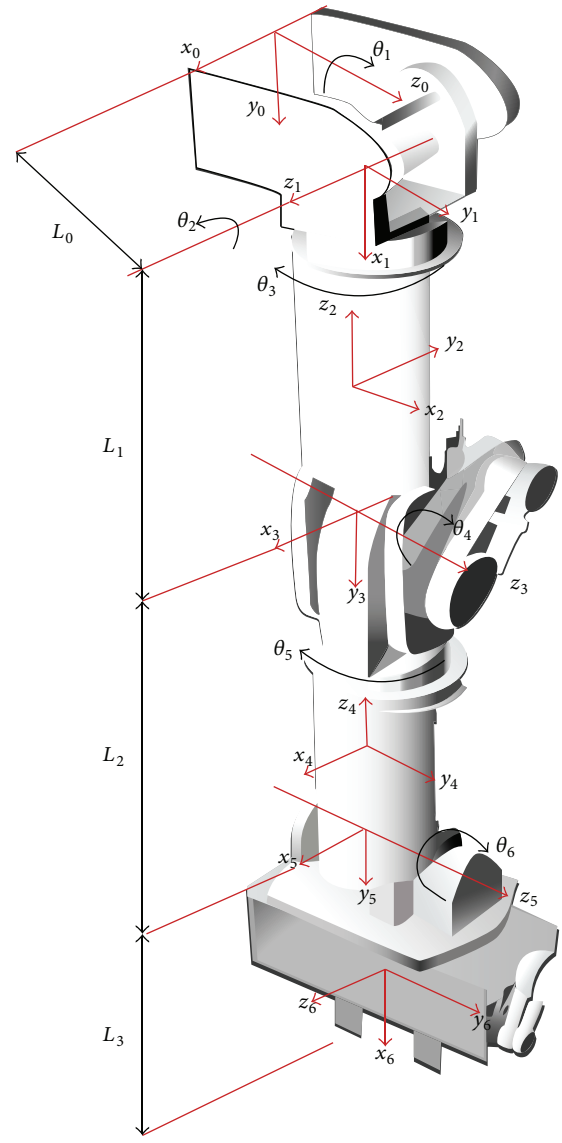


Figure 1: Picture of the general six-DOF humanoid robot arm.

In the forward kinematics equation (3), there are infinite inverse kinematics solutions for the six-DOF humanoid robot arm. Although the solution of the forward kinematics problem is steady forward, the solution of the inverse kinematics problem strictly depend on the robot arm's structures. It is difficult to obtain a general closed-form and optimal solution by using conventional methods, such as analytical approach [7, 8] and Jacobian-based approach [9]. In other words, the complex and time-consuming procedure of inverse kinematics must be derived again for different humanoid robot robots using these conventional approaches.

In order to circumvent this disadvantage, this paper presents an evolutionary DNA swarm intelligence algorithm combined with Taguchi method to solve the general six-DOF redundant inverse kinematics problem, thus determining the optimal joint variables of the six-DOF humanoid robot arm. The DNA parameters are determined using Taguchi theory rather than trial-and-error approach. The proposed DNA algorithm with Taguchi method evolves the optimal configuration of the joint variables for the six-DOF humanoid robot arm by minimizing the predefined fitness function and signal-to-noise ratio (SNR).

3. Taguchi-Based DNA Inverse Kinematics Redundancy Solver

3.1. DNA Swarm Intelligence Algorithm. DNA swarm intelligence algorithm is a subcategory of nature-inspired optimization and population-based optimization algorithms using the biomolecular structure of DNA molecules [17–20]. Individuals (chromosomes) cooperate with each other to find an optimal solution in the problem space. The basic elements of biological DNA are nucleotides which can be classified into four bases: adenine (A), guanine (G), cytosine (C), and thymine (T). A triplet code of nucleotide bases specifies the codon, which in turn contains a specific anticodon on transfer RNA (tRNA) and assists subsequent transmission of genetic information in the formation of a specific amino acid [17–20]. A chromosome consists of combinations of the above four bases with different chemical structures and can represent many genes in the optimization problems.

Numerical DNA computing algorithm is similar to GAs since their own natural genetic operators help evolution of the genes generation by generation, such as crossover and mutation; however it is different from GAs. In particular, the DNA algorithms provide two new operators, enzyme and virus, which are very useful to enhance the effect of mutation. Moreover, the coding scheme of DNAs is quite different from those of GAs. In what follows the coding scheme and core operators of DNA swarm intelligence algorithms are summarized.

3.1.1. Coding Scheme. A single strand of DNA can be represented by a string consisting of a combination of four different symbols, A, G, C, and T; that is, DNA algorithms use A, G, T, and C to stand for their chromosomes; for example, one can define A = 0, G = 1, T = 2, and C = 3 to encode the chromosomes. Compared with the conventional GAs in which the binary coding system is employed, this approach has more genetic information for complex optimization problems solving. Moreover, (4) can be easily applied to define the range and precision of a parameter in the DNA algorithm [17, 18]:

$$\pi = \frac{U_{\max} - U_{\min}}{4^l - 1}, \tag{4}$$

where π denotes the precision, l stands for how many bits will be used, $U_{\max}$ is the maximum of the parameter, and $U_{\min}$ denotes the minimum of the parameter. It is worth mentioning that (4) hinges on the fact that DNA swarm intelligence algorithms use four bits for the coding scheme, but GAs use only two bits.

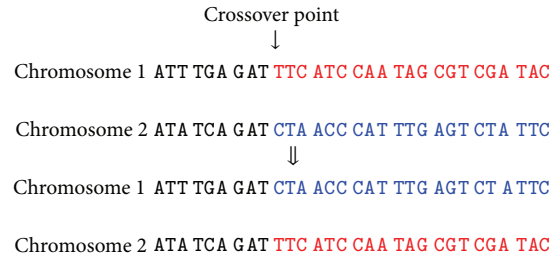


Figure 2: Illustration of one-point crossover of the chromosomes.

3.1.2. Selection (Reproduction). The main task of selection module is to select individuals from the populations so that these individuals can be sent to the crossover and mutation module in order to attain new offsprings. Selection is one of the key operators that ensure survival of the fitness. There are several selection methods with different characteristics, such as roulette selection, rank selection, and tournament selection. However, it is necessary to get the fitness value of each chromosome prior to selection process.

3.1.3. Crossover. Crossover is the main procedure of chromosomal exchange in the DNA swarm intelligence algorithm to achieve an effective database of knowledge. In molecular biology, crossover is the process by two chromosomes, paired up during prophase of meiosis, which exchanges some distal portion of their DNA. Crossover occurs when two chromosomes break and then reconnect but to different end piece. If they break at the same place, then the result is an exchange of genes, called genetic recombination. One-point crossover is adopted throughout the paper due to high speed operation shown in Figure 2 with a crossover rate P_c.

3.1.4. Mutation (Enzyme and Virus). Mutation is the process which consists of making small alterations to the bits of the chromosomes by applying some kind of randomized changes, such as single-point or multipoint mutation process. This operation complements the procedures of crossover and selection. As shown in Figure 3, in DNA swarm intelligence algorithms, there are two special mutation operators, enzyme and virus, which are more effective than GAs. The enzyme

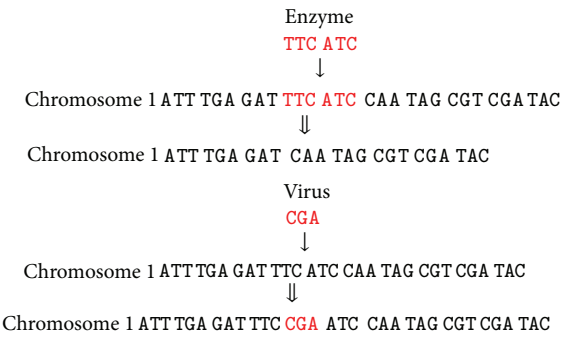


Figure 3: Illustration of enzyme and virus operations in DNA algorithm.

operator refers to deletion, in which one or more base pairs are removed, while the virus operator refers to insertion, in which one or more base pairs are inserted into the sequence. These two operations provide continuous renewal of the population, the searching diversity is increased, and the premature convergence problem is avoided. The determination of whether mutation will occur is decided by the mutation rate P_m.

3.1.5. Fitness Function (Performance Index). The fitness function is application-specific and is always designed according to the problem to be optimized. The fitness of new chromosomes from genetic operations, such as crossover and mutation, should be evaluated based on the fitness function. For complex problems, the computation time becomes dominant in the overall performance.

3.2. DNA Computing for Solving the Inverse Kinematics Problem. In DNA computing for inverse kinematics, each chromosome contains the set of the joint variables $[\theta_1 \quad \theta_2 \quad \theta_3 \quad \theta_4 \quad \theta_5 \quad \theta_6]$ of the six-DOF humanoid robot arm. The optimal solution is evolved by using the DNA evolution process to solve the redundant problem of the humanoid robot arm, thereby obtaining the optimal configurations $q = [\theta_1 \quad \theta_2 \quad \theta_3 \quad \theta_4 \quad \theta_5 \quad \theta_6]^T$ for the desired position of the end-effector $p = [p_x \quad p_y \quad p_z]^T$.

In the inverse kinematics redundant problem of the six-DOF humanoid robot arm, there exist infinitely many postures for the robotic arm from any starting pose to any destination pose. The proposed DNA algorithm will be employed to minimize the whole movement of the robot arm in the redundant problem. The configuration of the robot arm can be represented by the following vector matrix:

$$q = [\theta_1 \quad \theta_2 \quad \theta_3 \quad \theta_4 \quad \theta_5 \quad \theta_6]^T. \tag{5}$$

To determine the minimal movement of the robot arm, one defines a fitness function composed of the differences from the initial states of manipulator and the final states of the robot arm. The initial states q_i of the robot arm are denoted by

$$q_i = [\theta_{1,i} \quad \theta_{2,i} \quad \theta_{3,i} \quad \theta_{4,i} \quad \theta_{5,i} \quad \theta_{6,i}]^T. \tag{6}$$

The final states of the humanoid robot arm are expressed by

$$q_f = [\theta_{1,f} \quad \theta_{2,f} \quad \theta_{3,f} \quad \theta_{4,f} \quad \theta_{5,f} \quad \theta_{6,f}]^T. \tag{7}$$

The desired final position of the robot's end-effector is represented as $X_{m,f} = [x_{m,f} \quad y_{m,f} \quad z_{m,f}]^T = [p_x \quad p_y \quad p_z]^T$. Based on the desired final position of the end-effector and the inverse kinematics of the robotic arm in Section 2, one can obtain all the joint angles of the humanoid robotic arm using the DNA optimization process.

Fitness function (performance index) is very important in the DNA evolution process because it evaluates all the DNA chromosomes. In order to employ the DNA algorithm to solve the redundant problem, one defines the fitness function F as follows by considering the minimal movement of the robot arm:

$$\begin{aligned} F &= F_{\text{displacement}} + F_{\text{error}} \\ &= \left[(q_f - q_i)^T (q_f - q_i)\right]^{1/2} (q_f - q_i)^T \left(q_f - q_i\right) \\ &\quad + \omega \left[(x_{m,f} - x_{m,i})^2 + (y_{m,f} - y_{m,i})^2 + (z_{m,f} - z_{m,i})^2\right]^{1/2}, \end{aligned} \tag{8}$$

where $X_{m,i} = [x_{m,i} \quad y_{m,i} \quad z_{m,i}]^T$ is the initial position of the end-effector of the robot arm and ω is the weight factor. The fitness function F combines the position error F_{error} with an additional term based on the joint angle displacement $F_{\text{displacement}}$ from the initial position. The proposed DNA algorithm is then adopted to find the optimal solutions of the fitness function F in (7); that is, the best fitness value is thus obtained and the optimal configuration is determined. These optimal parameters, including $\theta_1, \theta_2, \theta_3, \theta_4, \theta_5$, and θ_6, are required for the design of robot arm controller to perform the desired tasks.

The DNA swarm intelligence algorithm for searching the optimal configurations of the humanoid robot arm is described by the following steps. Note that if the corresponding joint angle is within the limit of the motor's workspace, the chromosome is acceptable.

Step 1. Initialize the population size S, crossover rate P_c, mutation rate P_m, and number of iterations N.

Step 2. Set the two parents from the selection operation.

Step 3. Execute the crossover operation and also check whether new chromosomes are acceptable. If the new chromosomes are unacceptable, repeat this procedure until acceptable chromosomes are obtained.

Step 4. Perform the mutation process with low mutation rate and ensure that new chromosomes are acceptable.

Step 5. Repeat the previous four steps again until the convergence criterion is met or predetermined number of iterations is reached. Output the optimal configuration of the humanoid robot arm $q^* = [\theta_1^* \quad \theta_2^* \quad \theta_3^* \quad \theta_4^* \quad \theta_5^* \quad \theta_6^*]^T$.

Table 2: Control factors and their levels.

Control factor	Level 1	Level 2	Level 3
S	100	200	300
N	50	75	100
P_c	0.6	0.7	0.8
P_m	0.015	0.03	0.05

3.3. Taguchi DNA Inverse Kinematics Solver. Although the redundant inverse kinematics problem of the six-DOF humanoid robot arm has been successfully resolved by the proposed DNA swarm intelligence, the DNA parameters, such as population size, number of generation, crossover rate, and mutation rate, are not appropriately set. These parameters affect the searching performance of the DNA computing in inverse kinematics solver. The most important problem is, therefore, how to find the optimal DNA parameters to increase the search efficiency and decrease the effect of randomness in the proposed DNA redundancy solver. This subsection aims at employing Taguchi quality method to present a Taguchi-based DNA inverse kinematics solver for the six-DOF humanoid robot arm.

Taguchi experimental method can reduce the disturbance caused by randomness and can help determine the optimal parameters in the proposed DNA algorithm. The advantage of the combination of the DNA computing and Taguchi method is that only a few experiments are needed because of the orthogonal array of the Taguchi method. The proper DNA parameters can be set without full-parameter experiments. This approach not only greatly reduces the number of iterations but also makes the DNA final solution approaches the optimal values with less randomness. Taguchi method was constructed based on the principle of an orthogonal array that can effectively minimize the number of experiments required in any design process [21–25]. An orthogonal array is a fractional factorial matrix that provides a balanced comparison of levels of parameters or factors. This approach can provide an efficient way to obtain the optimal parameters or factors in an optimization problem or system design.

In this study, there are four control factors: population size (S), number of generation (N), crossover rate (P_c), and mutation rate (P_m). Table 2 shows the control factors and their levels, each factor has three levels (level 1, level 2, and level 3). Table 3 is a standard Taguchi orthogonal array $L_9(3^4)$ which is applied in this inverse kinematics solver. The full factorial design method requires $3^4 = 81$ experiments while the Taguchi method needs only nine experiments to obtain the approximate optimal values.

Signal-to-noise ratio (SNR) is used in Taguchi method to measure the quality of each experiment in the orthogonal array. There are several SNRs available depending on the type of characteristic: nominal-is-best, smaller-the-better, or larger-the-better [21–25]. This paper adopts the smaller-the-better characteristic and the SNR is given by

$$\text{SNR} = -10 \log F^2, \tag{9}$$

where F is the fitness value in (8).

Table 3: Orthogonal array of the inverse kinematics solver.

Exp. number	S	N	P_c	P_m
1	100	50	0.6	0.015
2	100	75	0.7	0.03
3	100	100	0.8	0.05
4	200	50	0.7	0.05
5	200	75	0.8	0.03
6	200	100	0.6	0.015
7	300	50	0.8	0.03
8	300	75	0.6	0.015
9	300	100	0.7	0.05

With the Taguchi method with orthogonal array, the parameters in DNA swam intelligence are properly set to obtain optimal performance by only using a few experiments. This Taguchi-based DNA (TDNA) solver provides a general redundancy resolution for six-DOF the humanoid robot arm. The proposed TDNA solver can be easily extended to resolve the redundancy problem of seven-DOF humanoid robot arms. The proposed TDNA solver outperforms the conventional solvers, such as geometric solver, Jacobian-based solver, GA-based solver, and ACO-based solver, because more genetic information and Taguchi quality method are applied in this natural-inspired swarm intelligence.

Remark 1. Once the optimal configuration is obtained via the TDNA algorithm, the next goal is to plan the motion profile of each joint and the time trajectory is then generated between the initial and final configurations. This motion trajectory planning can be achieved via the trapezoid velocity profile [26] and S-curve profile which are commonly used in motion control to generate the time trajectories for the joint angles and avoid strong random motion.

Remark 2. The proposed TDNA solver is easily extended to address the full inverse kinematics in which the position and rotation matrix are considered. Moreover, this efficient TDNA solver is applicable to any multi-DOF robot arms with different structures.

4. Simulation Results and Discussion

The aims of the simulations are to examine the effectiveness and performance of the proposed Taguchi-based DNA algorithm to solve the redundant inverse kinematics problem of the six-DOF humanoid robot arm. The initial states $q_i = [q_{m,i}] = [0^\circ \ 0^\circ \ 0^\circ \ 0^\circ \ 0^\circ \ 0^\circ]^T$ and the desired final position of the end-effector is set (20 cm, 20 cm, 10 cm). The Taguchi-based DNA computing is terminated at 100th generation. The evolved optimal joint solution is $(\theta_{1,f}, \theta_{2,f}, \theta_{3,f}, \theta_{4,f}, \theta_{5,f}, \theta_{6,f}) = (76.19^\circ, -79.15^\circ, -43.46^\circ, -68.17^\circ, 28.68^\circ, 1.06^\circ)$ and its corresponding position is (19.98 cm, 19.84 cm, 9.98 cm) which is very close to the desired position.

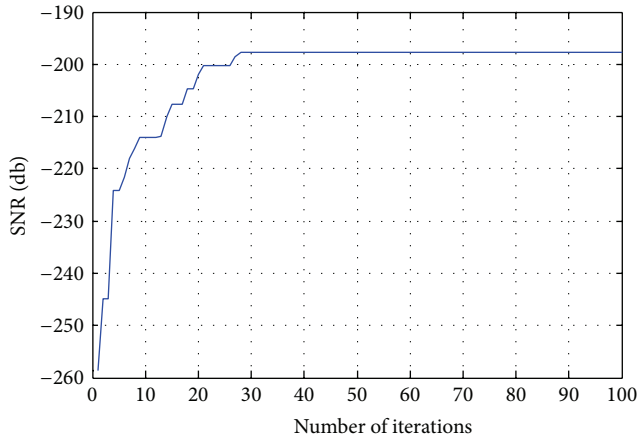


Figure 4: SNR of the proposed Taguchi-based DNA algorithm for the humanoid robot arm.

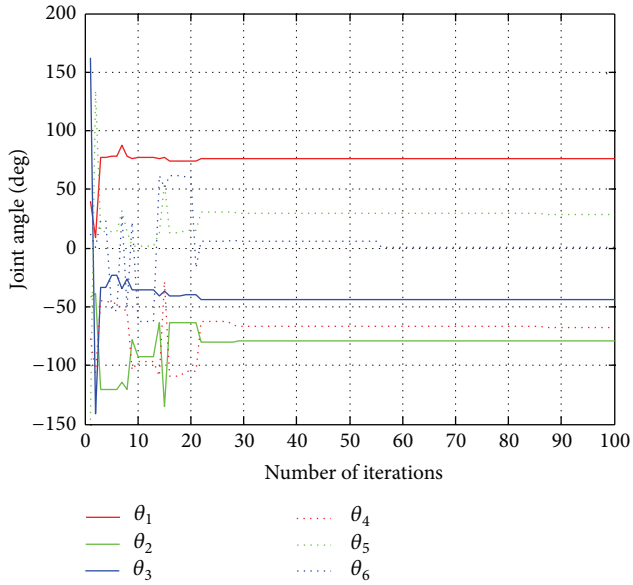


Figure 5: Joint trajectories of the robot arm moving from the starting point to the ending point.

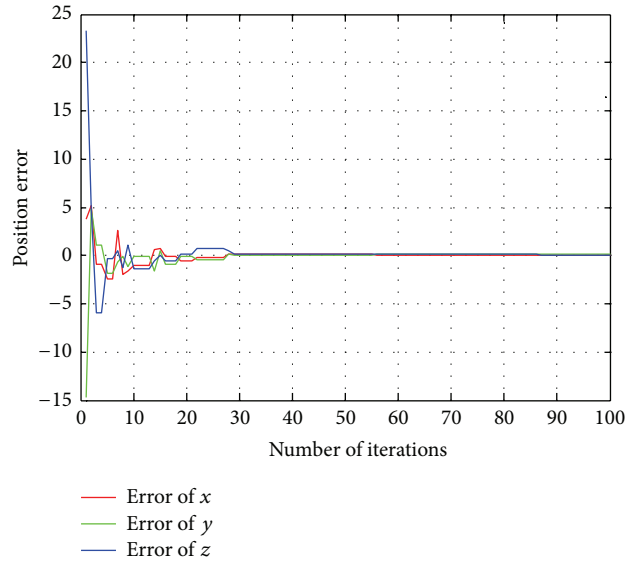


Figure 6: Position error of the humanoid robot arm.

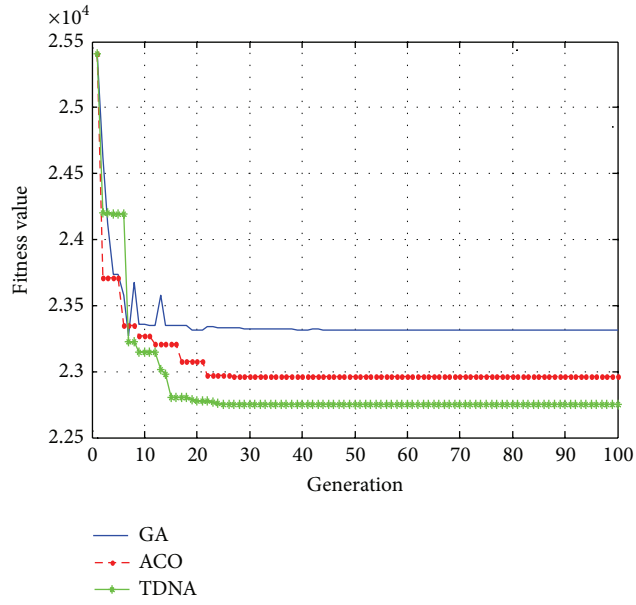


Figure 7: The evolutions of performance index in the proposed TDNA and conventional solvers.

Figure 4 presents the SNR of the proposed Taguchi-based DNA algorithm to solve the redundant problem of the six-DOF robot arm. As can been seen in Figure 4, the proposed Taguchi-based DNA algorithm successfully searches for the optimal configurations of the humanoid robot arm. Figure 5 depicts the joint trajectories of the robot manipulator moving from the starting point to the ending point. All the joint angles $\theta_1 \sim \theta_6$ converge to constant values. Figure 6 presents the convergence of end-effector's position error. As shown in Figure 6, the position errors, including x, y, and z, are nearly converged to zero successfully. These simulations clearly indicate that the proposed Taguchi-based DNA resolves the redundant problem and finds optimal configuration of the six-DOF robot arm.

In order to exhibit the merit of the proposed TDNA over conventional GA and ACO to solve the redundancy problem of the humanoid robot arm, Figure 7 presents the evolutions of performance index (fitness value) for the proposed TDNA solver and the two conventional solvers. The GA and ACO were executed with the parameters settings: population size of individuals is 100, crossover probability is 0.85, mutation probability is 0.1, and maximum iterations are 100, $\alpha = 0.2$, $\beta = 0.3$, $\rho = 0.7$, and $Q = 10$. These three solvers employed the same fitness function in (8) to cope with the same inverse kinematics redundancy problem of the six-DOF humanoid robot arm. As shown in Figure 7, the proposed TDNA solver converges to the optimal solution with better performance index. Through these results, the proposed TDNA inverse

kinematics solver is superior to the conventional solvers; that is, it solves this robotic optimization problem more efficiently with better joint configurations. It is worth mentioning that some metaheuristic algorithms could be considered in this area in the future, like swarm dolphin algorithm, computational intelligence aided design (CIAD) framework, and so on, which could be employed in future work.

5. Conclusions

This paper has presented an efficient inverse kinematics redundancy solver of six-DOF humanoid robot arm using Taguchi-based DNA swarm intelligence. The proposed hybrid algorithm has been developed to solve the redundancy problem. This algorithm has more genetic information and avoids the premature convergence problem in conventional methods. Through simulation results, the proposed TDNA optimal solver has been shown to search for the optimal solutions successfully. This TDNA solver has been shown to outperform the conventional solvers, such as geometric solver, Jacobian-based solver, GA solver, and the ACO solver.

Conflict of Interests

The authors declare that there is no conflict of interests regarding the publication of this paper.

Acknowledgment

The authors gratefully acknowledge the financial support from the National Science Council, Taiwan, under Grant NSC 102-2221-E-197-023.

References

[1] E. S. Neo, K. Yokoi, S. Kajita, and K. Tanie, "Whole-body motion generation integrating operator's intention and robot's autonomy in controlling humanoid robots," *IEEE Transactions on Robotics*, vol. 23, no. 4, pp. 763–775, 2007.

[2] Q. Li, F. Liu, and L. Liang, "The application of adaptive backstepping sliding mode for hybrid humanoid robot arm trajectory tracking control," *Advances in Mechanical Engineering*, vol. 2014, Article ID 307985, 9 pages, 2014.

[3] J. Nassour, V. Huge, F. B. Ouezdou, and G. Cheng, "Qualitative adaptive reward learning with success failure maps: applied to humanoid robot walking," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 24, no. 1, pp. 81–93, 2013.

[4] D. S. Zhang, R. Xiong, J. Wu, and J. Chu, "Balance maintenance in high- speed motion of humanoid robot a rm-based on the 6D constraints of momentum change rate," *The Scientific World Journal*, vol. 2014, Article ID 535294, 13 pages, 2014.

[5] J.-K. Yoo and J.-H. Kim, "Fuzzy integral-based gaze control architecture incorporated with modified-univector field-based navigation for humanoid robots," *IEEE Transactions on Systems, Man, and Cybernetics B: Cybernetics*, vol. 42, no. 1, pp. 125–139, 2012.

[6] S. G. Qiao, Q. Z. Liao, S. M. Wei, and H. J. Su, "Inverse kinematic analysis of the general 6R serial manipulators based on double quaternions," *Mechanism and Machine Theory*, vol. 45, no. 2, pp. 193–199, 2010.

[7] J. Q. Gan, E. Oyama, E. M. Resales, and H. Hu, "A complete analytical solution to the inverse kinematics of the Pioneer 2 robotic arm," *Robotica*, vol. 23, no. 1, pp. 123–129, 2005.

[8] G. Antonelli, S. Chiaverini, and G. Fusco, "A new on-line algorithm for inverse kinematics of robot manipulators ensuring path tracking capability under joint limits," *IEEE Transactions on Robotics and Automation*, vol. 19, no. 1, pp. 162–167, 2003.

[9] J. Yan and H. Huang, "A fast and smooth walking pattern generator of biped robot using Jacobian inverse kinematics," in *Proceedings of the IEEE Workshop on Advanced Robotics and its Social Impacts (ARSO '07)*, December 2007.

[10] P. K. Patchaikani, L. Behera, and G. Prasad, "A single network adaptive critic-based redundancy resolution scheme for robot manipulators," *IEEE Transactions on Industrial Electronics*, vol. 59, no. 8, pp. 3241–3253, 2012.

[11] Y. Zhang, S. Wang, and G. Ji, "A rule-based model for bankruptcy prediction based on an improved genetic ant colony algorithm," *Mathematical Problems in Engineering*, vol. 2013, Article ID 753251, 10 pages, 2013.

[12] H. C. Huang, "Intelligent motion control for omnidirectional mobile robots using ant colony optimization," *Applied Artificial Intelligence*, vol. 27, no. 3, pp. 151–169, 2013.

[13] J. Chen, X. Chen, and W. Liu, "Complete inverse method using ant colony optimization algorithm for structural parameters and excitation identification from output only measurements," *Mathematical Problems in Engineering*, vol. 2014, Article ID 185487, 18 pages, 2014.

[14] H. C. Huang, "Intelligent motion control for four-wheeled holonomic mobile robots using FPGA-based artificial immune system algorithm," *Advances in Mechanical Engineering*, vol. 2013, Article ID 589510, 11 pages, 2013.

[15] S. W. Lin and S. C. Chen, "Parameter tuning, feature selection and weight assignment of features for case-based reasoning by artificial immune system," *Applied Soft Computing Journal*, vol. 11, no. 8, pp. 5042–5052, 2011.

[16] R.-J. Wai, J.-D. Lee, and K.-L. Chuang, "Real-time PID control strategy for maglev transportation system via particle swarm optimization," *IEEE Transactions on Industrial Electronics*, vol. 58, no. 2, pp. 629–646, 2011.

[17] L. M. Adleman, "Molecular computation of solutions to combinatorial problems," *Science*, vol. 266, no. 5187, pp. 1021–1024, 1994.

[18] C. L. Lin, H. Y. Jan, and T. H. Huang, "Self-organizing PID control design based on DNA computing method," in *Proceedings of the IEEE International Conference on Control Applications*, pp. 568–573, September 2004.

[19] M. Karakose and U. Cigdem, "QPso-based adaptive DNA computing algorithm," *The Scientific World Journal*, vol. 2013, Article ID 160687, 8 pages, 2013.

[20] H. Jiao, Y. Zhong, and L. Zhang, "Artificial DNA computing-based spectral encoding and matching algorithm for hyperspectral remote sensing data," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 50, no. 10, pp. 4085–4104, 2012.

[21] G. Taguchi, *Introduction to Quality Engineering, Quality Resources*, Quality Resources, White Plains, NY, USA, 1986.

[22] Y. Su, Z. Bao, F. Wang, and T. Watanabe, "Efficient GA approach combined with Taguchi method for mixed constrained circuit design," in *Proceedings of the International Conference on Computational Science and Its Applications (ICCSA '11)*, pp. 290–293, Santander, Spain, June 2011.

[23] C. D. Yang, C. C. Luo, S. J. Liu, and Y. H. Chang, "Applications of genetic-Taguchi algorithm in flight control designs," *Journal of Aerospace Engineering*, vol. 18, no. 4, pp. 232–241, 2005.

[24] D. Seenivasan, V. Selladurai, and P. Senthil, "Optimization of liquid desiccant dehumidifier performance using Taguchi method," *Advances in Mechanical Engineering*, vol. 2014, Article ID 506487, 6 pages, 2014.

[25] H. M. Hasanien, "esign optimization of PID controller in automatic voltage regulator system using Taguchi combined genetic algorithm method," *IEEE Systems Journal*, vol. 7, no. 4, pp. 825–831, 2013.

[26] Y. Kung and G. Shu, "Design and implementation of a control IC for vertical articulated robot arm using SOPC technology," in *Proceedings of the IEEE International Conference on Mechatronics (ICM '05)*, pp. 532–536, Taipei, Taiwan, July 2005.

Hindawi Publishing Corporation
Mathematical Problems in Engineering
Volume 2014, Article ID 753487, 10 pages
http://dx.doi.org/10.1155/2014/753487

Research Article

OL-DEC-MDP Model for Multiagent Online Scheduling with a Time-Dependent Probability of Success

Cheng Zhu, Jiangfeng Luo, Weiming Zhang, and Zhong Liu

Science and Technology on Information Systems Engineering Laboratory, National University of Defense Technology, Changsha 410073, China

Correspondence should be addressed to Jiangfeng Luo; nudtluojiangfeng@gmail.com

Received 11 February 2014; Revised 5 June 2014; Accepted 1 July 2014; Published 22 July 2014

Academic Editor: Yun Li

Focusing on the on-line multiagent scheduling problem, this paper considers the time-dependent probability of success and processing duration and proposes an OL-DEC-MDP (opportunity loss-decentralized Markov Decision Processes) model to include opportunity loss into scheduling decision to improve overall performance. The success probability of job processing as well as the process duration is dependent on the time at which the processing is started. The probability of completing the assigned job by an agent would be higher when the process is started earlier, but the opportunity loss could also be high due to the longer engaging duration. As a result, OL-DEC-MDP model introduces a reward function considering the opportunity loss, which is estimated based on the prediction of the upcoming jobs by a sampling method on the job arrival. Heuristic strategies are introduced in computing the best starting time for an incoming job by each agent, and an incoming job will always be scheduled to the agent with the highest reward among all agents with their best starting policies. The simulation experiments show that the OL-DEC-MDP model will improve the overall scheduling performance compared with models not considering opportunity loss in heavy-loading environment.

1. Introduction

Problems involving time-dependent success probability extensively exist in manufacturing, industrial, and military domains. One example is the scheduling of a procrastinator [1], whose speed and success probability of job processing will increase as the due date is approaching. Practice shows that a procrastinator's performance varies under different time pressures when processing the same job. As higher time pressure is more likely to force a procrastinator to make mistakes when processing a sophisticated job, the success probability is consequentially dependent on the starting time. Another example is the antiship missile defense by SAM (surface-air-missile) systems shown in [2]. SAM systems are scheduled to intercept the incoming antiship missiles within feasible interception time window. Killing probability of the interception is associated with the range at which the interception missile and the antiship missile meets, which in turn depends on the launching time of the interception. Usually, an earlier firing time means more flight time before engagement.

Both of the above examples imply that though early starting strategy for job processing guarantees maximal time window for processing, longer processing duration will be spent as a price. Compared with the classic on-line scheduling [3, 4], extra trade-offs should be considered by the agent between the current job and the possible incoming jobs. For example, Figure 1 gives the killing probability associated with the time at which the engagement occurs for a SAM system of a Halifax ship against the incoming antiship missile [5]. It can be inferred that an antiship missile can be intercepted in the feasible time window $[t^l, t^u]$. Hence the SAM system can choose the best engaging time t_m to get the highest killing probability. If the interception fails, SAM system will have time window $[t_m, t^u]$ to take an immediate remedial interception. However, if the SAM system fires at earlier time and makes the engagement occur at t^l, though the killing probability is lowered down, longer time window is left in case of interception fail. Therefore, the SAM system needs to make trade-off between a high killing probability of current interception and more feasible time left to take remedial

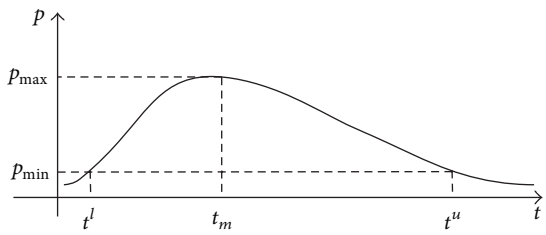


Figure 1: Engagement time-dependent killing probability of the interception missile.

action in case of interception fail. In addition, for an antiship missile and a SAM system, an earlier firing time always means more flight time before the engagement. Hence adopting the early firing strategy would cause SAM system to spend longer duration on the current interception, while losing more opportunities to intercept possible upcoming missiles. This is the second trade-off to be considered.

Similar trade-off also exists in the application of procrastinator on-line scheduling [1]. To the best of our knowledge, multiagent on-line scheduling with the trade-off discussed above is not studied by previous researches such as time-dependent scheduling [6–9], on-line stochastic optimization [3, 4, 10, 11], and stochastic resource allocation in a multiagent environment [12–15]. Therefore, in this paper, we consider the above trade-offs in a multiagent scheduling process. There are several independent agents that can be scheduled to process the stochastically arriving jobs. Each job has a specific feasible time window, during which an agent can process it with time-dependent success probability. In case of fail, an agent will immediately make another try as long as the remaining time window allows. The objective is to complete all jobs with high probability. A general problem definition is introduced in Section 2, and Section 3 surveys closely related studies. Section 4 builds a DEC-MDP (decentralized Markov Decision Processes) to model the on-line multiagent scheduling process without considering the opportunity loss. An OL-DEC-MDP (opportunity loss-decentralized Markov Decision Processes) model is proposed in Section 5 to include the opportunity loss in the scheduling decision with proofs on its properties. Section 6 is the simulation evaluation of the OL-DEC-MDP, and Section 7 contains the conclusions and the future work.

2. Problem Definitions

There is a group of agents, denoted as I, that should be scheduled to process a set of stochastically arriving jobs, which is denoted as J. For each job $j \in J$, there is a time interval $T_j = [t_j^l, t_j^u]$, during which the process of job j is feasible. For example, t_j^l and t_j^u are the low bound and upper bound of the feasible time window for job j to be processed. For each agent $i \in I$, there is a duration $d_{ij}(t)$, which should be spent to process job j for one time starting from time t. The outcome of the process by the end of $d_{ij}(t)$ is either success or fail, and probability of success is denoted as $p_{ij}(t)$.

Assumption 1. One agent can only be scheduled to process one job at a time.

Assumption 2. If the agent fails to complete the job by the end of the process, it will immediately start another try as long as the feasible processing time window of the job will not elapse before the next try can be finished.

Assumption 3. The agent will be released from the current job and be available for the next job, if either the current job is completed successfully or the current job is discarded because of insufficient time window left for another try.

According to the above assumptions, an agent i will have several opportunities to complete a job j before the feasible time window $[t_j^l, t_j^u]$ of job j elapses depending on the process duration $d_{ij}(t)$ of each try. For example, If a process starting from time t_0 fails by the end of time $t_0 + d_{ij}(t_0)$, agent i must try to reprocess the job immediately at time $t_0 + d_{ij}(t_0)$, as long as $t_0 + d_{ij}(t_0) + d_{ij}(t_0 + d_{ij}(t_0)) \le t_j^u$.

Assumption 4. For an assignment of job j to agent i, later a try of process starts; later the process will end, but the process duration will be shorter.

Assumption 4 is in accordance with the observations in the time-sensitive applications such as air defense. With the threat approaching, the time needed for an interception is diminishing. For example, the earlier a process starts, the earlier the effect can be observed, but a longer duration needs to be spent. According to Assumption 4, For any t_1, $t_2 \in [t_j^l, t_j^u]$, s.t. $t_1 + d_{ij}(t_1) \in [t_j^l, t_j^u]$, $t_2 + d_{ij}(t_2) \in [t_j^l, t_j^u]$, if $t_1 < t_2$, then $d_{ij}(t_1) > d_{ij}(t_2)$ and $t_1 + d_{ij}(t_1) < t_2 + d_{ij}(t_2)$.

Assumption 5. Each agent operates independently, and there is no resource competition or mutual influence between agents.

The objective is to schedule the agents on-line to successfully complete all the arriving jobs with highest probability. In the off-line case, the objective of the problem can be modelled as (1)

$$\max \prod_{i \in I} \left(1 - \prod_{j \in J} \left(1 - p_{ij}(t)\right)^{u_{ij}(t)} \right) \tag{1}$$

s.t.

$$u_{ij}(t) = \begin{cases} 1 & \text{if agent } i \text{ starts to process} \\ & \text{or reprocess job } j \text{ at time } t \\ 0 & \text{otherwise,} \end{cases} \tag{2}$$

$$\sum u_{ij}(t) \ge 0,$$

$$t \in \left[t_j^l, t_j^u\right], \quad t + d_{ij} \in \left[t_j^l, t_j^u\right], \quad \text{for any } u_{ij}(t) = 1; \tag{3}$$

objective function (1) is to maximize the probability of successfully completing all the incoming jobs. Constraint (2) implies that an agent can process a job more than one time.

Constraint (3) ensures that a try of job processing should not be started if the feasible time window of the job will elapse before a try can be finished.

Compared with a similar model in Karasakal et al. [2], the starting time of job processing in the above model is continuously distributed in a job's feasible time window. Moreover, in the on-line version, the future arriving jobs could not be known in advance, which makes the model more difficult to solve.

As a result, trade-off should be made during the on-line scheduling of the problem to ensure good scheduling quality:

(1) the trade-off between the probability of successful process of the current job and the probability of the successful reprocess of the job in case of fail;

(2) the trade-off between the reward of successful process of the current job and the opportunity loss that the agent might have with other incoming jobs during the current job processing.

3. Related Works

3.1. Scheduling. Job scheduling [16] is a classic domain to solve the problem, in which jobs need to be handled by one ore more machines regarding the constraints of due date, processing time, priorities, and so forth. There are many different models such as single or parallel machine model depending on the number of machines. If a job needs to be handled by a series of machines in-order, the models are called flow shops, job shops, or open shops under different situations. The objective is to handle all the jobs with a minimum makespan [17] or lateness [18, 19]. Time-independent uncertainties such as machine breakdowns, unexpected releases of jobs with high priority [16], duration of a processing [20], and execution uncertainty [21] are introduced in the scheduling, which are called stochastic scheduling. Recently, models on time-dependent scheduling are proposed, in which parameters of the scheduling are time-dependent. For example, learning effect and processing time are defined as increasing funtion [22, 23] or deceasing function [1, 24, 25] of their start times. However, most of the above studies are discussed in the off-line case, where all of the jobs exist from the beginning. Moreover, time-dependent parameters mainly focus on the processing times or the cost of processing, while time-dependent success probability of job processing is not discussed.

3.2. On-Line Stochastic Optimization. On-line stochastic optimization, such as the on-line packet scheduling, stochastic reservations, vehicle dispatching, or routing, has been studied [3, 4], in which a job or requisition arrives stochastically in queue to wait for a certain machine or a sever to be served. A job or requisition will be successfully processed once a machine is scheduled. Scheduling can be centralized or decentralized depending on whether the scheduling decision is made globally or by each agent. To model real-world problems, time-independent uncertainties such as action duration [11], resource consumption [10], and operation outcomes [12, 14] are introduced into the scheduling process. Sampling approach is also introduced to estimate the future arriving jobs to achieve a global optimal solution [26–28]. Similar to our problem, each agent of the above problem is dynamically scheduled against the stochastically arriving jobs, and there is no resource competition or dependences among the agents. However, time-dependent probability of success is not considered, and each job will be successfully processed once an agent is scheduled to process it. Moreover, job discarding is allowed if a more important job is arriving. Instead, in our problem, each agent should retry in case of fail as long as the time allows.

3.3. Stochastic Resource Allocation in Multiagent Environment. The most related studies in the area of stochastic resource allocation in multiagent environment mainly focus on the following problems; each agent can execute a task independently while different agents may share the same resources. An agent consuming shared resources may decrease the reward of other agents. As the outcome of the job execution is uncertain, the resources are allocated to achieve the global optimal solution. [12] solves this type of problems by introducing dynamic constraint satisfaction problem (DCSP) model into MDP and constructing a Markovian CSP (MaCSP) model. The best action at each Markovian step depends on the resource availability. As the state space increases exponentially with the number of agent and the types of resource, some studies propose heuristic search [29] and decomposition approaches [14, 15] in solving Decentralized Markov Decision Processes (DEC-MDP). As the dependency between different agents is taken into account, starting an action too early or too late by an agent may jeopardize the operation of others. Hence, trade-off is introduced into DEC-MDP to estimate the cost that one agent may suffer due to the negative influence of others [13].

Comparing between our problems with closely related studies is listed in Table 1.

4. Decentralize MDP (DEC-MDP)

In theory, models (1)–(3) can obtain the optimal scheduling solution for an off-line problem. However, in the on-line case, the scheduling decision should be made according to the state of each agent and the incoming job in real time. As a result, MDP provides a suitable approach to model the on-line scheduling by mapping the current state of agents and incoming jobs to an optimal scheduling decision. In order to construct the MDP model of a problem, the state space of the problem should be defined.

4.1. States of the Agent and the Job. The state of an agent is either *busy* when it is processing a job or *unemployed* when it is released from the current job. Let $s_a^i(t)$ denote the state of agent i at time t:

$$s_a^i(t) = \begin{cases} \text{busy} & \text{if agent } i \text{ is processing a job} \\ & \text{at time } t \\ \text{unemployed} & \text{otherwise.} \end{cases} \quad (4)$$

Table 1: Comparing of the closely related works.

	Stochastic jobs or tasks	Multiagent (machine)	Dependences or resource competition among agents	Time-dependent processing time or action duration	Due date for jobs	On-line sampling	DEC-MDP	Time-dependent probability of success
Time-dependent scheduling		√		√	√			
On-line stochastic optimization	√	√			√	√		
Stochastic resource allocation of multiagent		√	√		√		√	
OL-DEC-MDP model based on-line scheduling	√	√		√	√	√	√	√

If a job is being processed by an agent, the state of the job is modelled as the ratio of the remaining time window feasible for the job to be completed. If it is waiting to be processed, its state is set to be 0; if it has been completed successfully, the state is set to be 1; otherwise the state is set to be −1. Let $s_m^j(t)$ denote the state of job j at time t:

$$s_m^j(t) = \begin{cases} 0 & \text{if } t < t_j^l \text{ or } j \text{ has not been scheduled to any agent;} \\ \dfrac{t - t_j^l}{t_j^u - t_j^l} & \text{if } t \in \left[t_j^l, t_j^u\right), \text{ and } j \text{ is being processed by an agent} \\ 1 & \text{if } t \geq t_j^l \text{ or } j \text{ has been completed;} \\ -1 & \text{if } j \text{ has been discarded.} \end{cases} \tag{5}$$

For example, at time 0s, there is an unemployed agent i without any job coming. The state of the agent i at 0s is

$$s_a^i(0) = \text{unemployed}. \tag{6}$$

At time 10s, a job j arrives with feasible time window [12, 30], and it is scheduled to agent i which is due to start at time 12s. Then

$$\begin{gathered} s_a^i(t) = \text{unemployed}, \quad s_m^j(t) = 0 \quad \text{for } t \in [10, 12) \\ s_a^i(t) = \text{busy}, \qquad s_m^j(t) = \frac{t - 12}{30 - 12} \\ \text{for } t \in \left[12, 12 + d_{ij}(12)\right). \end{gathered} \tag{7}$$

By Assumption 4, there is $12 + d_{ij}(12) \leq 30$. Suppose $d_{ij}(12) = 8$, and job processing ends successfully; then

$$s_a^i(20) = \text{unemployed}, \qquad s_m^j(20) = 1. \tag{8}$$

If job processing fails by the time 20s, agent i will start another try to process job j immediately as long as $20 + d_{ij}(20) \leq 30$ holds. Suppose $d_{ij}(20) = 9$; then

$$\begin{gathered} s_a^i(t) = \text{busy}, \qquad s_m^j(t) = \frac{t - 12}{30 - 12} \\ \text{for } t \in \left[20, 20 + d_{ij}(20)\right). \end{gathered} \tag{9}$$

If job processing fails again by the end of the second try (e.g., at time 29s), and the remaining time window of job j (only 1 second is left, since $t_j^u = 30$) is not enough for another try, then job j will be discarded from time 29s:

$$s_a^i(t) = \text{unemployed}, \quad s_m^j(t) = -1 \quad \text{for } t > 29. \tag{10}$$

In this case, agent i will be available for other incoming jobs from time 29s. As the different agents may be released or start to process a job at different times, it is hard to define a joint action, which is the set of actions for each agent in each decision step of the on-line scheduling process [13]. Moreover, because of the time-dependent state space, the reward of a joint action is difficult to evaluate by a recursive approach as introduced in [30]. Recently, in order to limit the set of state space in the multiagent environment, there is significant progress in extending the Markov Decision Processes (MDP) for optimizing decentralized control [13, 31]. In this paper, as there is no dependence or resource competition among agents, a decentralized MDP is adopted to model the decision process of each agent.

4.2. DEC-MDP. For an agent i and its allocated job j with time window $[t_j^l, t_j^u]$, the corresponding DEC-MDP is defined as a tuple $\langle s_{ij}, A_{ij}, p_{ij}, r_{ij}\rangle_{t_1}$, where

(i) $s_{ij} = (s_a^i, s_m^j)$ is the state set regarding agent i and job j during the whole process period (e.g., from the time

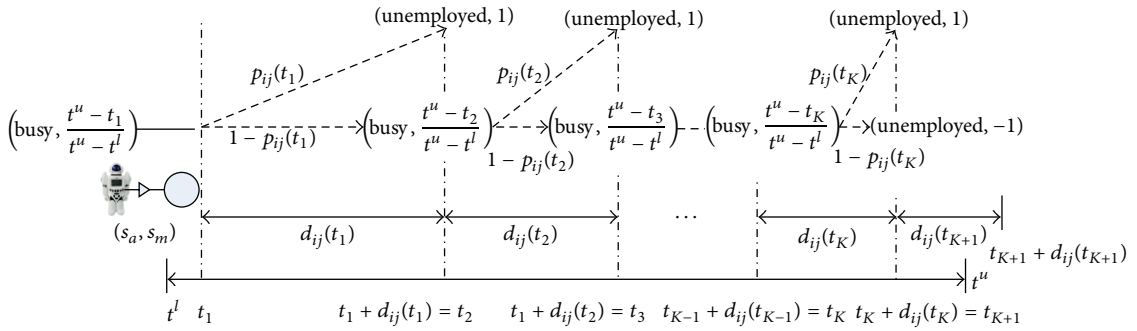


Figure 2: The state transition process of MDP $\langle s_{ij}, A_{ij}, p_{ij}, r_{ij}\rangle_{t_1}$.

when the process is started for the first time, denoted as t_1, to the time when agent i is released from job j);

(ii) A_{ij} is the strategy set of agent a_i, represented by the starting time that agent i may choose to process job j for the first time. $A_{ij} = \{t \mid t \in [t_j^l, t_j^u], t + d_{ij}(t) \in [t_j^l, t_j^u]\}$;

(iii) $p_{ij}(t)$ is a function of time t, which gives the success probability of completing job j by agent i by the time of $t + d_{ij}(t)$, when the process or the reprocess starts at time t. According to Assumptions 2 and 3, there is $p_{ij}(t) = 0$ if $t \notin [t_j^l, t_j^u]$ or $t + d_{ij}(t) \notin [t_j^l, t_j^u]$;

(iv) $r_{ij}(t)$ is the reward function as defined in (13).

The initial state of a DEC-MDP is

$$s_{ij}(t_1) = \left\{ s_a^i(t_1) = \text{busy}, s_m^j(t_1) = \frac{\left(t_1 - t_j^l\right)}{\left(t_j^u - t_j^l\right)} \right\}. \quad (11)$$

The absorbing state is

$$s_{ij}(\cdot) = \left\{ s_a^i(\cdot) = \text{unemployed}, s_m^j(\cdot) = 1 \text{ or } -1 \right\}. \quad (12)$$

Figure 2 shows the state transition process when agent a_i starts to process job j at time t_1 for the first time, in which the maximal retry times are K, and K is the smallest number that satisfies $t_{K+1} + d_{ij}(t_{K+1}) > t_j^u$. The reward function is defined as in (13), which is represented by the probability that agent i will successfully complete job j when starting the first process at time t_1. Consider

$$r_{ij}(t_1) = p_{ij}(t_1) + \sum_{x=2}^{K} \left(\prod_{y=1}^{x-1} \left(1 - p_{ij}\left(t_y\right)\right) \right) \cdot p_{ij}\left(t_x\right) \quad (13)$$

s.t.

$$t_x \in \left[t_j^l, t_j^u\right], \quad t_{x+1} = t_x + d_{ij}\left(t_x\right), \quad t_{x+1} \in \left[t_j^l, t_j^u\right], \quad x = 1, 2, \ldots, K \quad (14)$$

$$t_{K+1} + d_{ij}\left(t_{K+1}\right) > t_j^u; \quad (15)$$

when starting from time t_1, agent i can process job j not more than K times. With (13)–(15), the best time for agent i to begin to process job j is t_1^*:

$$t_1^* = \arg\max_{t \in A_{ij}} r_{ij}(t). \quad (16)$$

Therefore, during the on-line scheduling, we prefer scheduling the incoming job j to an agent i^* with highest success probability:

$$i^* = \arg\max_{i \in I} r_{ij}\left(t_1^*\right). \quad (17)$$

However, as stated in Section 4, this decision does not take the opportunity loss into account. Agent may lose higher rewards with upcoming jobs during its engagement with the current job. As a result, we introduce a opportunity loss decentralized MDP (OL-DEC-MDP) model in the next section.

5. Opportunity Loss Decentralized MDP (OL-DEC-MDP)

An OL-DEC-MDP model has the same state space, strategy set, and transition probability with a DEC-MDP. However, the reward function of an OL-DEC-MDP should be redefined to take the opportunity loss into account.

5.1. Opportunity Loss. As shown in Figure 2, the agent may try at most K times before being released from the current job. It will not be available for other upcoming jobs during the period $\Delta T_{ij}^z(t_1) = [t_1, t_z + d_{ij}(t_z)]$ $(1 \le z \le K)$ with the probability

$$p_{ij}^z = \begin{cases} 1 & z = 1 \\ \prod_{x=1}^{z-1} \left(1 - p_{ij}\left(t_x\right)\right) & 2 \le z \le K, \end{cases} \quad (18)$$

where $t_{x+1} = t_x + d_{ij}(t_x)$ and $x = 1, 2, \ldots, z-1$. As a result, agent will lose all upcoming jobs during $\Delta T_{ij}^z(t_1)$ with probability of p_{ij}^z. Hence the opportunity loss for agent i to process job j starting from t_1 can be defined as

$$\text{OL}(i, j, t_1) = \max_{1 \le z \le K} \left(p_{ij}^z \max_{q \in M\left(\Delta T_{ij}^z(t_1)\right)} r_{iq}\left(t_1^q\right) \right). \quad (19)$$

In the above equation, t_1^q is the best starting time for agent i to process job q, which is decided by (16). $M(\Delta T_{ij}^z(t_1))$ is the set of all possible jobs that will arrive during period $\Delta T_{ij}^z(t_1)$. The opportunity loss of the agent is defined to be the highest possible reward that the agent may lose during the period of engagement with its current job. Considering both the reward and the potential loss in scheduling decision, we now refine the reward function in OL-DEC-MDP as following:

$$r'(i, j, t_1) = r_{ij}(t_1) - \mathrm{OL}(i, j, t_1). \quad (20)$$

As a result, the best starting time of agent i to process job j is t_1^*:

$$t_1^* = \arg\max_{t \in A_{ij}} r'(i, j, t). \quad (21)$$

Reward function (21) calculates the maximum reward when schedule agent i to process job j while taking the opportunity loss into account.

5.2. Computation of the Reward with Opportunity Loss. For an OL-DEC-MDP $\langle s_{ij}, A_{ij}, p_{ij}, r'_{ij}\rangle_{t_1}$, if job j is allocated to agent i, then agent i should choose a starting time t_1 from the time window $[t_j^l, t_e]$, while $t_e + d_{ij}(t_e) = t_j^u$. If derivation of the reward function (21) exists within interval $[t_j^l, t_e]$, the optimal starting time t_1^* can be decided as following:

$$\frac{\partial\left[r_{ij}(t_1) - \mathrm{OL}(i, j, t_1)\right]}{\partial t_1} = 0 \quad \left(t_1 \in \left[t_j^l, t_e\right]\right). \quad (22)$$

However, if derivation of reward function (21) does not exist within interval $[t_j^l, t_e]$, t_1 can be decided with the following heuristics.

(1) Start as early as possible. For example, set $t_1 = t_j^l$ if the agent is available earlier than t_j^l, or set t_1 to be the earliest time when agent i becomes available. We denote t_1 under this heuristic as t_a, $t_a \in [t_j^l, t_e]$.

(2) Start as late as possible, as long as agent will still have the same number of retrying opportunities (K) with starting as early as possible (e.g., at time t_a). With this heuristic, there are $t_{x+1} = t_x + d_{ij}(t_x)$ $(x = 1, 2, \ldots, K-1)$ and $t_K + d_{ij}(t_K) = t_j^u$. We denote t_1 under this heuristic as t'.

(3) Start at the time with highest success probability to complete job j within the first try, while still having the maximal retrying opportunities (K). For example, t_1 is the time point between $[t_a, t']$ with the highest success probability of the first try. With this heuristic, there is $t_1 = \arg\max_{t \in [t_a, t']} p_{ij}(t)$. We denote t_1 under this heuristic as t_m.

(4) Start at the time with the highest success probability to complete the job j within the first try. With this heuristic, there is $t_1 = \arg\max_{t \in [t_a, t_e]} p_{ij}(t)$, while $t_e + d_{ij}(t_e) = t_j^u$. We denote t_1 under this heuristic as t_M.

As a result, the best starting time t_1^* of agent i to process job j considering both rewards and opportunity loss can be computed as

$$t_1^* = \arg\max_{t_1 \in \{t_a, t', t_m, t_M\}} \left[r_{ij}(t_1) - \mathrm{OL}(i, j, t_1)\right]. \quad (23)$$

According to (19), to compute $\mathrm{OL}(i, j, t_1)$, we should know $M(\Delta T_{ij}^K(t_1))$; for example, the set of all possible jobs that will arrive during time $\Delta T_{ij}^K(t_1)$. $M(\Delta T_{ij}^K(t_1))$ can be estimated on-line by the sampling approach, as described in [26, 27], which can forecast the possible events according to the job arriving distribution.

5.3. On-Line Scheduling Based on OL-DEC-MDP. The detailed scheduling algorithm is given as following.

(1) Queuing Up the Incoming Jobs. When a new job comes, it is queued up in a time-priority queue. A new arrived job with a smaller low bound of feasible time will have higher priority.

(2) Observing the System State Change. Each agent has a job list with length of 1, which indicates its next job to be processed. System state changes when

(a) agent is released from current job and starts to process the assigned job in its job list (the agent's job list will be empty);

(b) agent fails to complete the current job and begins to make another try (the job in the agent's job list will be still waiting, which will be rescheduled);

(c) a new job is coming, and there exists at least one agent with empty job list (the incoming job will be assigned to some agent by being pushed into its job list).

(3) Scheduling/Rescheduling When System State Changes. When system state changes, scheduling or rescheduling decision will be made to decide or adjust the best next job as well as the best starting time for each agent. J_L denote the current set of next job of all agents before rescheduling.

(a) If $\|J_L\| < \|I\|$, then dequeue $\|I\| - \|J_L\|$ jobs from queue. Let J_n be the set of these dequeued jobs. Then, $J_L \cup J_n$ is the job set to be scheduled/rescheduled, as denoted by J_s.

(b) Order jobs in J_s according to time priority, and clear the job list of all agents.

(c) Schedule each job by order in J_s to an agent as following.

(i) Given job j, it will be scheduled to the agent i^* with highest reward:

$$i^* = \arg\max_{i \in I_A} r'(i, j, t_1^*). \quad (24)$$

In the above computation, I_A is the set of all agents with empty job list; t_1^* is decided by (23) with given i and j. For each agent, If it

has not been released when job j comes, its earliest available time t_a is set to be the ending time of its current process cycle. For example, the scheduling decision is made based on the assumption that all agents will be released by the end of its current process cycle. If the assumption is violated according to the observation, it is thought to be a system state change, and rescheduling will be made as described in step 3.

(ii) Push job j into job list of i^*.

(4) Job Processing. when available (being released from the current job), agent will begin to process the job in its job list at time t_1^*, and its job list will be cleared. By Assumption 2, agents will try many times to complete assigned jobs before success or time window of the job expires.

5.4. Properties and Proofs

Property 1. The time complexity to compute the best starting time t_1^* for an agent i to process job j according to (21) is $O(t_j^u - t_j^l)$.

Proof. As shown in Figure 2, there is one state for an OL-DEC-MDP in t_1, and two possible states in t_k, where $k = 2, 3, \ldots, K, K+1$. Hence, the maximum number of possible states during the job process is $2K+1$. As a result, the time complexity to calculate the expectation value of completing current job is $O(2K+1)$. On the other hand, the time complexity to calculate the maximal possible opportunity loss is $O(|M(\Delta T_{ij}^K)| \cdot (2K+1))$. Therefore, the time complexity to calculate the reward for a given agent i, job j, and start time t_1 is $O((|M(\Delta T_{ij}^K(t_1))|+1) \cdot (2K+1))$. As $|M(\Delta T_{ij}^K(t_1))|$ and K are constant for a given instance, we can set a constant $C = (|M(\Delta T_{ij}^K(t_1))|+1) \cdot (2K+1)$. Therefore, the time complexity to compute t_1^* in (21) is

$$O\left(\int_{t_j^l}^{t_j^u} C\right) = O\left(C \cdot \left(t_j^u - t_j^l\right)\right) = O\left(t_j^u - t_j^l\right). \quad (25)$$

□

Property 2. The average time for agent i on processing job j is $\sum_{z=1}^{K} p_{ij}^z \cdot (t_z + d_{ij}(t_z) - t_1)$, while $t_{z+1} = t_z + d_{ij}(t_z)$ and $z = 1, 2, \ldots, K-1$.

Proof. As shown in (17), given the starting time of the first processing t_1, the probability is p_{ij}^z for the agent i to spend $\Delta T_{ij}^z(t_1) = [t_1, t_z + d_{ij}(t_z)]$ on processing job j, where $t_{z+1} = t_z + d_{ij}(t_z)$ and $z = 1, 2, \ldots, K-1$. Thus, the average time for agent i to spend on job j is

$$\sum_{z=1}^{K} p_{ij}^z \cdot \left(t_z + d_{ij}\left(t_z\right) - t_1\right). \quad (26)$$

□

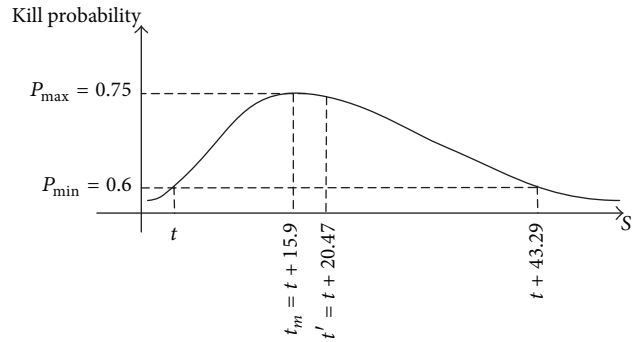


Figure 3: killing probability function.

6. Evaluations

6.1. Evaluation Setting. In the evaluation, a scenario of antiship missile defence by SAM systems is studied, which is introduced in [2].

Suppose there are four ship-borne SAM systems that can be scheduled to intercept the incoming antiship missiles, and each SAM system is capable of working independently and intercepting antiship missile coming from any direction (the modern ship-borne vertical launching missile system matches these features and is becoming very popular). The feasible interception time window of each incoming antiship missile j is set to be $[t_j^l, t_j + 43.29s]$, in which t_j^l is the time when the antiship missile is detected. The length of interception time window is decided by the detection capability of SAM system as well as the speed of the antiship missile. As a result, based on Section 5, if a SAM system is available when the missile is detected, there are $t_a = t_j^l$, $t_m = t_M = t_a + 15.9s$, and $t' = t_a + 20.47s$ based on scenario in [2]. The killing probability associated with the starting time of each interception $p_{ij}(t)$ is shown in Figure 3, which is approximated by a cubic multinomial. The duration function is defined as $d_{ij}(t) = r_{ij}(t)/(\nu_{\text{sam}} + \nu_m)$, where $r_{ij}(t)$ is the range at time t between the SAM missile and the antiship missile; ν_{sam} and ν_m are the velocities of the SAM missile and the antiship missile.

For the incoming antiship missile, the total number is set to be m ($m \in \{10, 12, 14, 16, 18\}$), and its arrival follows uniform distribution during a time span $[0, T]$ ($T \in \{20, 30, 40, 50, 60, 70, 80, 90\}$). To compute the opportunity loss as described in Section 5, a sampling method [26, 27] is implemented. Simulations are run under each combination of m and T to compare the scheduling result under different circumstances.

6.2. Quality of Scheduling. In the air-defence scenario, fail to intercept even one time may result in severe damage. Hence the quality of scheduling is measured by the probability of successfully interception of all incoming antiship missiles, which is denoted as *P-interception*. As shown in Figures 4 and 5, both DEC-MDP and OL-DEC-MDP based scheduling approaches illustrate that less intensive the attack comes (fewer antiship missiles with fixed time span or longer time

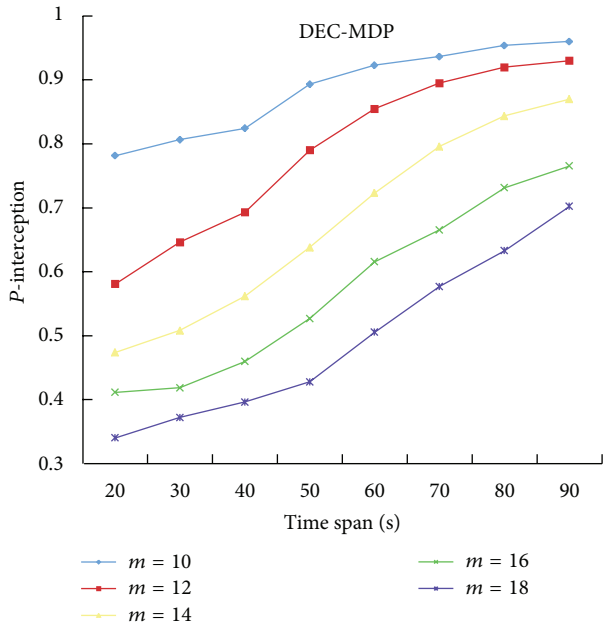


Figure 4: Probability of overall interception of DEC-MDP based approach.

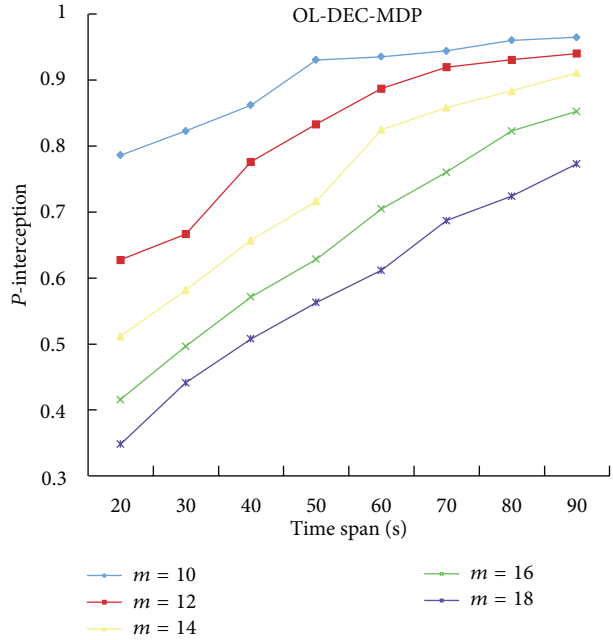


Figure 5: Probability of overall interception of OL-DEC-MDP based approach.

span with fixed incoming antiship missiles), higher the *P-interception* will be. The reason is that if there are fewer antiship missiles per time unit, there should be more available SAM systems that can be scheduled, hence the overall interception performance will be improved.

However, as shown in Figure 6, the OL-DEC-MDP based scheduling approach always has a higher probability of overall interception compared with DEC-MDP model. It can be observed that, for the same time span, the improvement of OL-DEC-MDP becomes more significant as the number of incoming missiles (m) increases. For example, there is performance improvement under more intensive attack environment. On the other hand, the overall shape of the improvement along the time span (for the fixed number of incoming antiship missiles, longer time span means less intensive attack) tends out to be a "cap." For example, the improvement rises sharply with the time span increasing at first and then comes down after reaching some peaks. The reason is that when the time span is small at first, which means that the antiship missiles are coming very intensively, it is very hard to improve the interception performance by OL-DEC-MDP since the SAM system reaches its saturation point under very intensive attack. The decision space left for each SAM system to decide the best starting time of the interception is quite small; hence OL-DEC-MDP has similar performance with DEC-MDP. However, when the intensity falls below the saturation point of the SAM system, the improvement brought by OL-DEC-MDP becomes gradually significant as opportunity loss is taken into account in on-line scheduling to achieve better overall performance. As the attack intensity continues to lower down with the increase of time span, the whole system has more than enough capability (available SAM systems) to intercept the incoming missiles; hence the improvement brought by OL-DEC-MDP becomes less significant.

Figure 7 shows the best starting time of interception used in OL-DEC-MDP obtained by heuristic strategy introduced in Section 5. Y-axis is the time indicating how long after an antiship missile is detected the first interception is launched. It can be observed that when the antiship missiles arrive intensively (which means smaller time span with fixed total incoming missiles), OL-DEC-MDP prefers to postpone the first interception launch. Study on the simulation data shows that the optimal starting time for interception under this case is near the time t_m, which means that the strategy to achieve the highest killing probability against the antiship missile by one shot is the superior strategy. This observation can be inferred from Property 2 that the superior strategy in this case is to release the SAM system as early as possible to treat the next incoming antiship missile. On the other hand, when the attack is less intensive (which means longer time span with fixed total incoming missiles), OL-DEC-MDP prefers to start the interception earlier as to leave more feasible time for retrying in case of interception fail.

7. Conclusions

This paper proposes an OL-DEC-MDP model for on-line multiagent stochastic scheduling, which considers the starting time-dependent probability of success and processing duration. The probability of completing the assigned job by an agent would be higher when the process is started earlier, but the opportunity loss could be also high due to the longer engaging duration. As a result, OL-DEC-MDP model introduces the reward function considering the opportunity

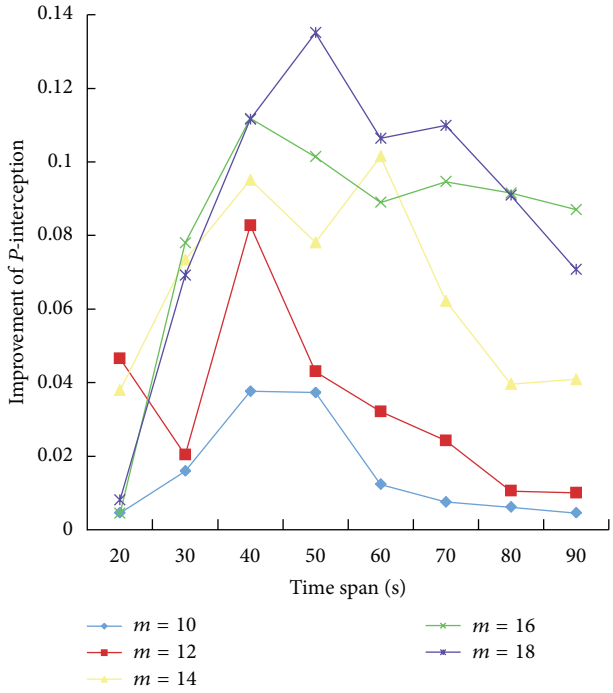


Figure 6: The increased probability of interception by OL-DEC-MDP compared with DEC-MDP.

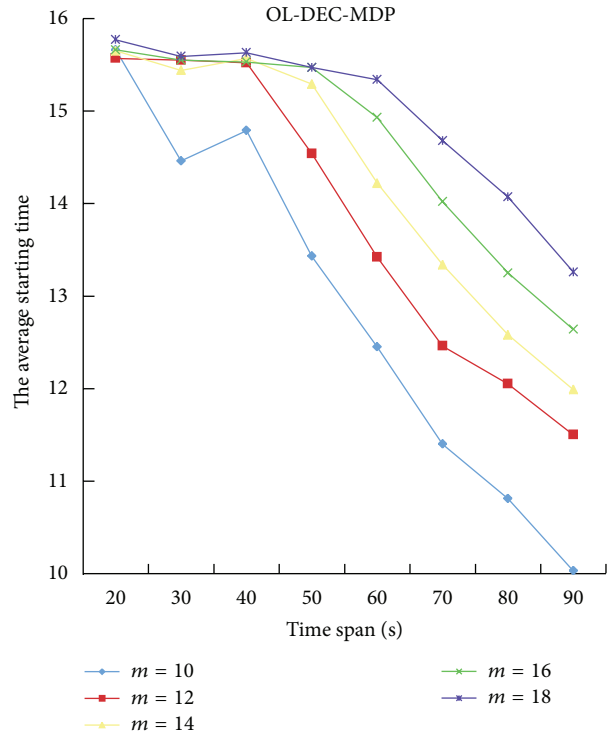


Figure 7: The average starting time of interception (t_1^*) computed by OL-DEC-MDP.

loss and schedules the incoming job to the agent with the highest reward. In order to measure the opportunity loss, OL-DEC-MDP model uses sampling method to predict the upcoming jobs and introduces heuristic strategies to compute the best starting time of an agent against an incoming job. The simulation experiments show that the OL-DEC-MDP model will improve the overall scheduling performance compared with models without considering opportunity loss, such as DEC-MDP. The overall trend of performance improvement is studied under different scenarios, which shows that the performance improvement is most significant if the jobs are coming intensively but within the saturation point of the multiagent system.

For the future research, we should extend the model to more general cases.

(1) *Dependency Between Agents*. In some cases, agents may interfere with other's operation. For example, if soft weapons such as chaff rocket are used during the interception, there may be mutual interference between different air defence weapons: firing a chaff rocket may prevent the missile guiding radar of the SAM system from working normally. In future work, the mutual influence between agents will be considered in constructing available strategy set and computing action reward.

(2) *Partial Observation*. For some real-world problem, the result of the action can only be partially observed. For example, the result of interception by a SAM system may not be totally observed by other agents due to the limitation of sensing capability. Hence the reward and the opportunity loss should be reevaluated, and POMDP (partial observation MDP) based approach could be a good candidate.

(3) *On-line Learning*. The sampling approach implemented in OL-DEC-MDP is based on the prior knowledge of the arrival distribution of the incoming jobs. If the prior knowledge of the arrival distribution does not exist, the on-line learning method could be used to learn and predict the future incoming jobs.

Conflict of Interests

The authors declare that there is no conflict of interests regarding the publication of this paper.

Acknowledgments

This paper is supported by the NSF of China under Grant nos. 61273322 and 71001105.

References

[1] M. A. Bender, R. Clifford, and K. Tsichlas, "Scheduling algorithms for procrastinators," *Journal of Scheduling*, vol. 11, no. 2, pp. 95–104, 2008.

[2] O. Karasakal, N. E. Özdemirel, and L. Kandiller, "Anti-ship missile defense for a naval task group," *Naval Research Logistics*, vol. 58, no. 3, pp. 304–321, 2011.

[3] P. Van Hentenryck and R. Bent, *Online Stochastic Combinatorial Optimization*, The MIT Press, Cambridge, Mass, USA, 2006.

[4] P. van Hentenryck, R. Bent, and E. Upfal, "Online stochastic optimization under time constraints," *Annals of Operations Research*, vol. 177, pp. 151–183, 2010.

[5] A. R. Benaskeur, F. Kabanza, and E. Beaudry, "CORALS: a real-time planner for anti-air defense operations," *ACM Transactions on Intelligent Systems and Technology*, vol. 1, no. 2, Article ID 1869402, p. 13, 2010.

[6] M. Debczynski and S. Gawiejnowicz, "An exact algorithm and a heuristic for scheduling linearly deteriorating jobs with arbitrary precedence constraints and the maximum cost criterion," in *Proceedings of the Federated Conference on Computer Science and Information Systems (FedCSIS '12)*, pp. 401–405, September 2012.

[7] S. Gawiejnowicz, *Time-Dependent Scheduling*, Monographs in Theoretical Computer Science. An EATCS Series, Springer, Berlin, Germany, 2008.

[8] S. Gawiejnowicz and B. M. T. Lin, "Scheduling time-dependent jobs under mixed deterioration," *Applied Mathematics and Computation*, vol. 216, no. 2, pp. 438–447, 2010.

[9] S. Gawiejnowicz, W. Kurc, and L. Pankowska, "Equivalent time-dependent scheduling problems," *European Journal of Operational Research*, vol. 196, no. 3, pp. 919–929, 2009.

[10] C. Dance and A. A. Gaivoronski, "Stochastic optimization for real time service capacity allocation under random service demand," *Annals of Operations Research*, vol. 193, pp. 221–253, 2012.

[11] M. J. Sobel, J. G. Szmerekovsky, and V. Tilson, "Scheduling projects with stochastic activity duration to maximize expected net present value," *European Journal of Operational Research*, vol. 198, no. 3, pp. 697–705, 2009.

[12] C. Besse and B. Chaib-draa, "An efficient model for dynamic and constrained resource allocation problems," in *Proceedings of the 2nd International Workshop on Constraint Satisfaction Techniques for Planning and Scheduling Problems (COPLAS '07)*, 2007.

[13] A. Beynier and A. Mouaddib, "Solving efficiently decentralized MDPs with temporal and resource constraints," *Autonomous Agents and Multi-Agent Systems*, vol. 23, no. 3, pp. 486–539, 2011.

[14] P. Plamondon and B. Chaib-Draa, "Stochastic resource allocation in multiagent environments: an approach based on distributed q-values and bounded real-time dynamic programming," *International Journal on Artificial Intelligence Tools*, vol. 21, no. 1, Article ID 1250003, 25 pages, 2012.

[15] P. Plamondon, B. Chaib-Draa, and A. R. Benaskeur, "A real-time dynamic programming decomposition approach to resource allocation," in *Proceedings of the IEEE Information, Decision and Control (IDC '07)*, pp. 308–313, February 2007.

[16] M. L. Pinedo, *Scheduling: Theory, Algorithms, and Systems*, Springer, New York, NY, USA, 2012.

[17] K. Fleszar, C. Charalambous, and K. S. Hindi, "A variable neighborhood descent heuristic for the problem of makespan minimisation on unrelated parallel machines with setup times," *Journal of Intelligent Manufacturing*, vol. 23, no. 5, pp. 1949–1958, 2012.

[18] M. A. González, C. R. Vela, I. González-Rodríguez, and R. Varela, "Lateness minimization with Tabu search for job shop scheduling problem with sequence dependent setup times," *Journal of Intelligent Manufacturing*, vol. 24, no. 4, pp. 741–754, 2013.

[19] M. D. ToksarI and E. Güner, "Parallel machine scheduling problem to minimize the earliness/tardiness costs with learning effect and deteriorating jobs," *Journal of Intelligent Manufacturing*, vol. 21, no. 6, pp. 843–851, 2010.

[20] J. Huang, G. A. Süer, and S. B. R. Urs, "Genetic algorithm for rotary machine scheduling with dependent processing times," *Journal of Intelligent Manufacturing*, vol. 23, no. 5, pp. 1931–1948, 2012.

[21] R. Rasconi, A. Cesta, and N. Policella, "Validating scheduling approaches against executional uncertainty," *Journal of Intelligent Manufacturing*, vol. 21, no. 1, pp. 49–64, 2010.

[22] M. Hosseinabadi Farahani and L. Hosseini, "Minimizing cycle time in single machine scheduling with start time-dependent processing times," *International Journal of Advanced Manufacturing Technology*, vol. 64, no. 9–12, pp. 1479–1486, 2013.

[23] W.-C. Lee, Y. S. Lin, and C.-C. Wu, "A branch-and-bound and heuristic algorithm for the single-machine time-dependent scheduling problem," *International Journal of Advanced Manufacturing Technology*, vol. 47, no. 9–12, pp. 1217–1223, 2010.

[24] D. Okołowski and S. Gawiejnowicz, "Exact and heuristic algorithms for parallel-machine scheduling with DeJong's learning effect," *Computers and Industrial Engineering*, vol. 59, no. 2, pp. 272–279, 2010.

[25] X.-Y. Wang, M.-Z. Wang, and J.-B. Wang, "Flow shop scheduling to minimize makespan with decreasing time-dependent job processing times," *Computers & Industrial Engineering*, vol. 60, no. 4, pp. 840–844, 2011.

[26] E. Burns, J. Benton, W. Ruml, S. Yoon, and M. B. Do, "Anticipatory on-line planning," in *Proceedings of the 22nd International Conference on Automated Planning and Scheduling (ICAPS '12)*, pp. 333–337, June 2012.

[27] H. S. Chang, R. Givan, and E. K. P. Chong, "On-line scheduling via sampling," in *Proceedings of the Conference on Artificial Intelligence Planning and Scheduling (AIPS '00)*, pp. 62–71, 2000.

[28] L. Mercier and P. van Hentenryck, "An anytime multistep anticipatory algorithm for online stochastic combinatorial optimization," *Annals of Operations Research*, vol. 184, pp. 233–271, 2011.

[29] N. Meuleau, E. Benazera, R. I. Brafman, E. A. Hansen, and M. Mausam, "A heuristic search approach to planning with continuous resources in stochastic domains," *Journal of Artificial Intelligence Research*, vol. 34, no. 1, pp. 27–59, 2009.

[30] R. Bellman, "A markovian decision process," DTIC Technical Document, DTIC, 1957.

[31] A. Beynier and A. Mouaddib, "A polynomial algorithm for decentralized Markov decision processes with temporal constraints," in *Proceedings of the 4th International Conference on Autonomous Agents and Multi agent Systems (AAMAS '05)*, pp. 963–969, July 2005.

Hindawi Publishing Corporation
Mathematical Problems in Engineering
Volume 2014, Article ID 365417, 23 pages
http://dx.doi.org/10.1155/2014/365417

Research Article

Global Appearance Applied to Visual Map Building and Path Estimation Using Multiscale Analysis

Francisco Amorós,[1] Luis Payá,[1] Oscar Reinoso,[1] Walterio Mayol-Cuevas,[2] and Andrew Calway[2]

[1] *Department of Systems Engineering and Automation, Miguel Hernández University, Avenida de la Universidad s/n, 03202 Elche, Spain*
[2] *Department of Computer Science, University of Bristol, Woodland Road, Bristol BS8 1UB, UK*

Correspondence should be addressed to Francisco Amorós; famoros@umh.es

Received 7 March 2014; Revised 11 June 2014; Accepted 11 June 2014; Published 22 July 2014

Academic Editor: Yi Chen

In this work we present a topological map building and localization system for mobile robots based on global appearance of visual information. We include a comparison and analysis of global-appearance techniques applied to wide-angle scenes in retrieval tasks. Next, we define multiscale analysis, which permits improving the association between images and extracting topological distances. Then, a topological map-building algorithm is proposed. At first, the algorithm has information only of some isolated positions of the navigation area in the form of nodes. Each node is composed of a collection of images that covers the complete field of view from a certain position. The algorithm solves the node retrieval and estimates their spatial arrangement. With these aims, it uses the visual information captured along some routes that cover the navigation area. As a result, the algorithm builds a graph that reflects the distribution and adjacency relations between nodes (map). After the map building, we also propose a route path estimation system. This algorithm takes advantage of the multiscale analysis. The accuracy in the pose estimation is not reduced to the nodes locations but also to intermediate positions between them. The algorithms have been tested using two different databases captured in real indoor environments under dynamic conditions.

1. Introduction

The autonomous navigation of a mobile robot usually involves a minimal knowledge of the surrounding environment. Normally, that knowledge is used with the purpose of building an internal representation of the area in a map. Using the map and the current information that the robot receives from its sensors, it is possible to carry out the localization of the robot and also to simultaneously add new information to the map.

In the literature, we can find a wide variety of environment representations depending on the sensor used. In this way, it is possible to find examples that try to compute the position of the robot using GPS, laser, or wheel encoders as input information sensors. Among all the possibilities, vision systems have become common sensors for robot control due to the richness of the information they provide, their relative low weight and cost, and the variety of possible configurations. So that, we can find researches based on single standard cameras as [1], wide-angle cameras [2], stereo cameras [3], catadioptric systems that provide us with omnidirectional images [4], or an array of cameras arranged circularly to obtain a panoramic image [5]. In this work, we use a fish-eye single camera, due to the fact that they provide a wide-angle view of the environment and have lower cost than other visual systems.

In the great majority of real visual applications, it is not possible to work with the image information directly from the sensors, as the memory requirements and computational cost would make the process unfeasible. Taking this into account, it is necessary to find an alternative representation of images that contains as much information as possible with a reduced memory size. In this task, two main categories can be found: feature based and global-appearance based

descriptors. The first approach is based on the extraction and description of significant points or regions from the scene. In this sense, we find examples of the use of SIFT features [6, 7], SURF [8, 9], or Harris edge and corner detector [10] applied to localization and mapping tasks. On the other hand, global-appearance descriptors try to describe the scene as a whole, without the extraction of local features or regions. These techniques have a special interest in unstructured and changing environments where finding patterns to recognize the scene might be difficult. For example, Kröse et al. [11] demonstrate the robustness of PCA (principal component analysis) applied to image processing. Menegatti et al. [12] take advantage of the properties of the discrete fourier transform (DFT) applied to panoramic images in order to build descriptors of the scene. In [13], Kunttu et al. describe the behaviour of a descriptor based on Fourier transform and Wavelet filter in image retrieving tasks.

Regarding the representation of the map, three main approaches stand out: metric, topological, and hybrid techniques. Metric maps include information of distances with respect to a predefined coordinate system. These maps provide the position of the robot except for an uncertainty associated with the sensor error. However, they usually have high computational cost. As an example, it is possible to find examples based on a sonar sensor applied to robot navigation [14] and other approaches that solve the SLAM problem using a team of robots with a map represented by the three-dimensional position of visual landmarks [15].

In contrast, topological techniques use graph-based representations of the environment. The nodes correspond to different areas of the environment, whereas the edges represent the connectivity relationships between the nodes. In those maps there are no absolute distances. Since they constitute a simple and compact representation, time and memory requirements are generally lower than in metric maps. However, they contain enough information to allow the robot to navigate autonomously through the environment. Many different visual-based navigation systems can be found, as [16, 17] present. They both use an omnidirectional camera as input sensor and topological maps as a representation of structured indoor office environments. In these maps, the nodes correspond to images of the areas where the robot navigates. These images are described using PCA techniques in order to reduce the memory requirements. The navigation between nodes relies on a visual path following algorithm that extracts the edges of corridor walls. A similar system but using a single camera is developed in [18]. Nodes correspond to special locations where certain actions must be taken, such as a turning or a door crossing; meanwhile, the edges represent trajectories where visual servoing navigation can be carried out. Specifically, the visual servoing is based on the vanishing point to keep the trajectory of the robot centered in the corridor. Other examples of topological map indoor navigation can be found in [19], where the gradient orientation histogram of the image is used in order to describe the scenes. Štimec et al. [20] present an appearance-based method for path-based map learning in both indoor and outdoor environments. This system is based on clustering of PCA features extracted from panoramic images in order to obtain distinctive visual aspects. In [21] we can find a topological localization system in a home environment. They use a sonar sensor and a grid map matching in order to carry out the localization of a robot, dealing also with the kidnapping problem. In [22], another example of topological homing navigation system is presented. In the proposal, some information from an omnidirectional visual system, a 3D stereo vision system, and the odometry are combined to carry out mapping and localization tasks using a mobile robot. FAB-MAP [23, 24] is another well-known topological SLAM approach, based on SURF features extraction to describe the appearance of the images. This algorithm has been tested in large scale navigation environments. On the other hand, Bellotto et al. [25] describe a visual topological localization system for mobile robot that uses digital image zooming. This work is based on the appearance of omnidirectional images and includes an image matching algorithm that improves the image association by means of the use of digital zoom of the scenes.

At last, regarding hybrid techniques, they try to take advantage of both topological and metric proposals. Normally, hybrid maps use metric in order to build local maps of separated areas, whereas topological relations are used in order to create a general map. It is also possible to introduce the topological relations to carry out loop closures in metric maps. An example of hybrid SLAM algorithm is RatSLAM [26]. This technique integrates the internal odometry of the robot provided by wheel encoders and visual information, using a neural network in a biologically inspired fashion. In [27] we can find the joint use of FAB-MAP and RatSLAM.

In this paper we propose a framework for only-visual topological map building and localization. Our technique stands out because of the use of global-appearance techniques to describe the scenes and the application of a multiscale analysis of the visual information to estimate relative distances between images. The system is intended for autonomous robotic navigation in mainly indoor spaces, such as offices and industrial environments where topological navigation is suitable.

The map is represented as a graph. In this graph, the nodes are collections of 8 wide-angle images captured every 45 degrees, covering the complete field of view around their positions. The topological distance between nodes, which is estimated by means of the multiscale analysis, will be proportional to the actual distance between positions of the nodes in the real world.

We use the information of several routes of images acquired along the environment, which pass through the nodes, to carry out the map building. The map building system is able to recognize new nodes, find their orientation, their relative position, and connectivity using these routes of images. We use the multiscale analysis to obtain both an increase of correct matching of route images in the map database and also a measurement of the relative position of the compared scenes.

After the map building, we propose a route estimation algorithm, which takes also advantage of that multiscale analysis. In this case, this analysis is used to enhance

the localization of the robot, being able to locate the robot not only in the nodes positions but also in intermediate points. We also introduce a weighting function that improves the localization accuracy using the graph obtained during the map building.

In [28], we find an example of visual route navigation using visual information that tries to keep the input memory to a minimum. Following that idea, we include a study of the computational cost of image retrieving using different global-appearance descriptors and image sizes in order to minimize the time and memory requirements. This study will be used to select the descriptor and the features of the images in the map building and localization experiments.

The following glossary includes some of the terms used in the text.

(i) *Node*: collection of 8 images captured from the same position on the ground plane every 45° approximately, covering the complete field of view around that position.

(ii) *Map database or node database*: collection of the images of all the nodes.

(iii) *Map*: graph that represents the topological layout of the nodes.

(iv) *Map building*: process of finding the topological connection between nodes and their relative position.

(v) *Topological distance*: relative position between images or nodes in the map.

(vi) *Image distance (d)*: Euclidean distance between the descriptors of two images.

The contributions of this paper are the following.

(i) We propose the multiscale analysis, which allows us to determine the relative position of two images captured with approximately the same orientation using global-appearance descriptors.

(ii) The paper includes a map building algorithm. Our algorithm lies in finding the spatial distribution of a collection of nodes of images distributed along an area. The system provides a topological graph that represents the adjacency relations and layout of the nodes.

(iii) We develop a topological localization system that extends the possible current pose estimation of the robot not only to the node locations but also to intermediate positions, making use of the multiscale analysis. This system allows the robot to estimate its path during the navigation using only the visual information.

(iv) Finally, we offer an experimental validation of the map building and route path estimation algorithms using our own database, which contains information of two different areas.

The remainder of the work is structured as follows. Section 2 introduces the specifications of the images captured to carry out the experiments and the global appearance descriptors considered to describe the scenes. In Section 3 we compare and analyze the computational requirements and matching precision of different descriptors using several image resolutions. Section 4 introduces the multiscale analysis. Next section presents the algorithm developed to build the topological map. In Section 6, we explain a novel route path estimation algorithm. Section 7 includes the experimental results using our own database. Finally, a summary with the main contributions in this work is included.

2. Images Features and Descriptors

This section describes the main features of the images used in the experimental part and the techniques we have applied in order to obtain a descriptor that extracts the main information from the images based on their global appearance.

The images are captured using a fisheye lens camera. Specifically, we use the Hero2 camera of GoPro [29]. We choose this camera due to its wide-angle field of view (127°), its low weight, and relative low cost compared to other visual systems.

The goal of the image descriptors is to solve the problem of place recognition using the global appearance of the scenes, trying to keep the memory requirements and computational cost to a minimum. The descriptors based on the global appearance concentrate the visual information of the image working with it as a whole, avoiding the extraction of landmarks or local features. They have presented good results in visual navigation tasks. It is possible to find previous works comparing these techniques [30, 31] or using them in map building and localization [32]. These researches use omnidirectional vision sensors. However, we do not have knowledge of any work where these techniques had been applied to nonpanoramic images.

Due to the use of a fisheye lens, the images captured with our camera present a radial distortion that must be corrected. It would be impossible to obtain useful information from the distorted images using global appearance descriptors, since they are based on the spatial distribution and disposition of the elements in the scene. For that reason, we use the Matlab Toolbox *OCamCalib* [33] to calibrate the camera and compute the undistorted scenes from the original images.

We consider UV the coordinate system of the image and XYZ the coordinate system of the real world, which is situated in the focal point of the lens. We consider the UV directions aligned with XY. The coordinates x and y of a point P in the real world are proportional to the coordinates u and v of an image point:

$$\begin{bmatrix} x \\ y \end{bmatrix} = \alpha \cdot \begin{bmatrix} u \\ v \end{bmatrix}. \tag{1}$$

Therefore, the vector $\vec{P}$ can be defined as

$$\vec{P} = \begin{bmatrix} x \\ y \\ z \end{bmatrix} = \begin{bmatrix} \alpha \cdot u \\ \alpha \cdot v \\ f(u, v) \end{bmatrix}. \tag{2}$$

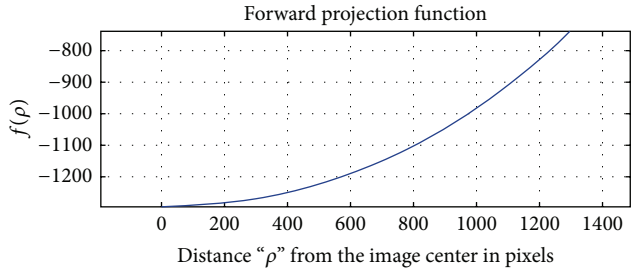


Figure 1: Forward projection function of the camera.

We can include the parameter α in the function $f(u, v)$. In this way, the previous equation can be expressed as

$$\vec{P} = \begin{bmatrix} x \\ y \\ z \end{bmatrix} = \begin{bmatrix} u \\ v \\ f(u, v) \end{bmatrix}. \quad (3)$$

Due to the symmetric geometry of the lens, the coordinate z of the point P only depends on the distance of the image point p regarding the image coordinate center:

$$\vec{P} = \begin{bmatrix} x \\ y \\ z \end{bmatrix} = \begin{bmatrix} u \\ v \\ f(\rho) \end{bmatrix}, \quad \text{with } \rho = \sqrt{u^2 + v^2}. \quad (4)$$

The function f, also named forward projection function, depends on the lens geometry. In general, it can be expressed as a n-degree polynomial:

$$f(\rho) = a_0 + a_1 \cdot \rho + a_2 \cdot \rho^2 + a_3 \cdot \rho^3 + \cdots + a_n \cdot \rho^n. \quad (5)$$

In our particular case, the minimum calibration error is obtained for a polynomial of degree equal to 3. The calibration function is

$$f(\rho) = -13194.89 + 2.252 \cdot 10^{-4} \rho^2 + 5.90 \cdot 10^{-8} \rho^3. \quad (6)$$

This function provides information about the direction of the rays that arrive to the camera system. The undistorted image corresponds to the protection of these rays in a plane parallel to the camera sensor.

Figure 1 shows the forward projection function of the camera, obtained after the calibration process.

In Figure 2 we can see an example of the original image and the undistorted view.

In the remainder of the paper, when we talk about the images, we will refer to the undistorted version of the original scenes.

Next, we include a summary of different descriptors based on the global appearance of the scenes.

2.1. Fourier-Based Techniques. It is possible to describe an image using the discrete Fourier transform over its rows. We can transform each row of the image $\{a_n\} = \{a_0, a_1, \ldots, a_{N-1}\}$ into the sequence of complex numbers $\{A_n\} = \{A_0, A_1, \ldots, A_{N-1}\}$:

$$\{A_n\} = \mathcal{F}[\{a_n\}] = \sum_{n=0}^{N-1} a_n e^{-j(2\pi/N)kn}; \quad k = 0, \ldots, N-1. \quad (7)$$

The most relevant information of the image is concentrated in the low frequencies. These frequencies represent large scale features of the images. Moreover, in real images, high frequencies are often affected by noise. Figure 3 shows the modules of the first components of the Fourier transform of each row of an image. Hence, we select only the first coefficients of the discrete Fourier transform of each row to build the descriptor.

In [12], Menegatti et al. present a descriptor that uses the discrete Fourier transform in panoramic images, defining the Fourier signature. Since the magnitude of the transform presents rotational invariance, in that case it constitutes the localization descriptor. However, our database images are not panoramic, and the rotational invariance may introduce localization errors in areas where there is a symmetry between different images, as corridors. To avoid it, our descriptor is not made up by the magnitude but by the original complex values.

2.2. Histogram of Oriented Gradients. The descriptor based on the histogram of oriented gradients (HOG) [34] uses the orientation of the gradient of an image. First we have to compute the spatial derivatives of the image along x and y-axis (I_x and I_y). Then, we obtain the magnitude and direction values of the gradient of each pixel:

$$|G| = \sqrt{I_x^2 + I_y^2}; \qquad \theta = \arctan\left(\frac{I_y}{I_x}\right). \quad (8)$$

Next, the image is divided into cells, and the histograms of the cells are computed. In Figure 4 we can see the division of the gradient of an image to obtain different cells (Figure 4(b)) and the estimation of the histogram of each cell (Figure 4(c)). The histogram is computed based on the gradient orientation of the pixels within the cell, weighted with the magnitudes of the gradient. The descriptor consists of the histograms' values of all the cell the image is divided into, ordered in a vector.

2.3. Gist-Based Techniques. Gist denotes a group of techniques that can be used to compress visual information as [35] details. These descriptors try to obtain the essential information of the images simulating the human perception system, that is, identifying a scene through its colour or remarkable structures, avoiding the representation of specific objects. Oliva and Torralba [36] develop this idea under the name of *holistic representation of the spatial envelope* to create a descriptor. In [37], this model uses global scene features, such as spatial frequencies and different scales based on Gabor filtering.

A Gabor filter is a lineal filter whose impulse response is a sinusoid modulated with a Gaussian function [38]. Therefore, a Gabor mask is localized both in the spatial and in the frequency domains (Figure 5). Thanks to its properties regarding textures treatment, Gabor filter can be used in compression and segmentation of digital images, as [39] shows.

First, we create a bank of Gabor masks including different resolutions and orientations. Then, the image is filtered using the set of filters. The orientation of each filtering depends

(a) (b)

Figure 2: (a) Image captured with a fisheye lens camera and (b) its corresponding undistorted view.

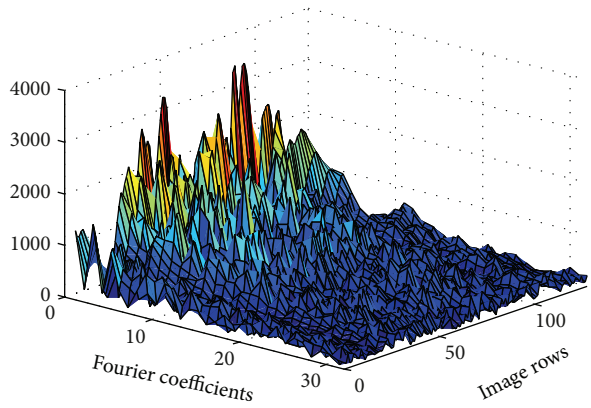


Figure 3: Module of the discrete Fourier transform of an image per rows.

on the number of masks of each resolution, since they are equally distributed between 0 and 180 degrees. The filtered images encode different structural information according to the mask applied. After that, the images are divided into cells, and we compute the average pixels' value within each cell. This process is represented in Figure 6. This is repeated for every filtered image. The final descriptor is composed of the mean value of intensity of the pixels contained in horizontal cells applied to every filtered scene.

3. Localization Recognition

In this section, we present a comparison between the different global appearance descriptors included in the previous section applied to location retrieval tasks. The aim of this study is to check the performance of these descriptors applied over perspective images using different resolutions. The comparison takes into account both the precision in correct matching and the computational requirements.

The database is composed of several nodes of images captured in different isolated places randomly chosen in both indoor and outdoor environments. Note that every node is composed of 8 images captured with a phase lag equal to 45 between consecutive images. We also take a set of test images.

Table 1: Image resolutions used in the experiments.

	Image's pixels
Size 1	1817 × 1004
Size 2	512 × 283
Size 3	256 × 128
Size 4	128 × 64
Size 5	64 × 32
Size 6	32 × 16
Size 7	16 × 8

The test images are captured in the same locations of the nodes with unknown orientations. Specifically, we capture 26 nodes and 3 test images per location. This database is different from the images used in the following sections.

In the experiments, we create a database with the descriptors of all the images of the nodes. When a new test image arrives, we compute its descriptor and compare it with the database. We define the image distance as the Euclidean distance between descriptors, which allows us to measure the similarity between scenes. Regarding the classifier, we choose the nearest neighbour.

Moreover, we are interested in finding the minimum image resolution we can use without detriment of matching precision. Table 1 shows the image sizes we have tested along the experiments. Size 1 is the original resolution of the camera.

In Figure 7 we can see the necessary time to compute the descriptor of an image. The information has been divided into two different graphs in order to clarify the data included. On the other hand, Figure 8 shows the memory requirements to store the image descriptors of all the nodes in a database. We can appreciate an exponential reduction of the requirements when we use smaller resolutions.

Gist-Gabor stands out as the computationally most expensive descriptor (Figure 7), although as the image size is reduced, the time differences between descriptors decrease. Regarding the memory requirements (Figure 8), the Fourier signature is the technique with higher memory requirements. Figure 9 shows the performance of each descriptor when

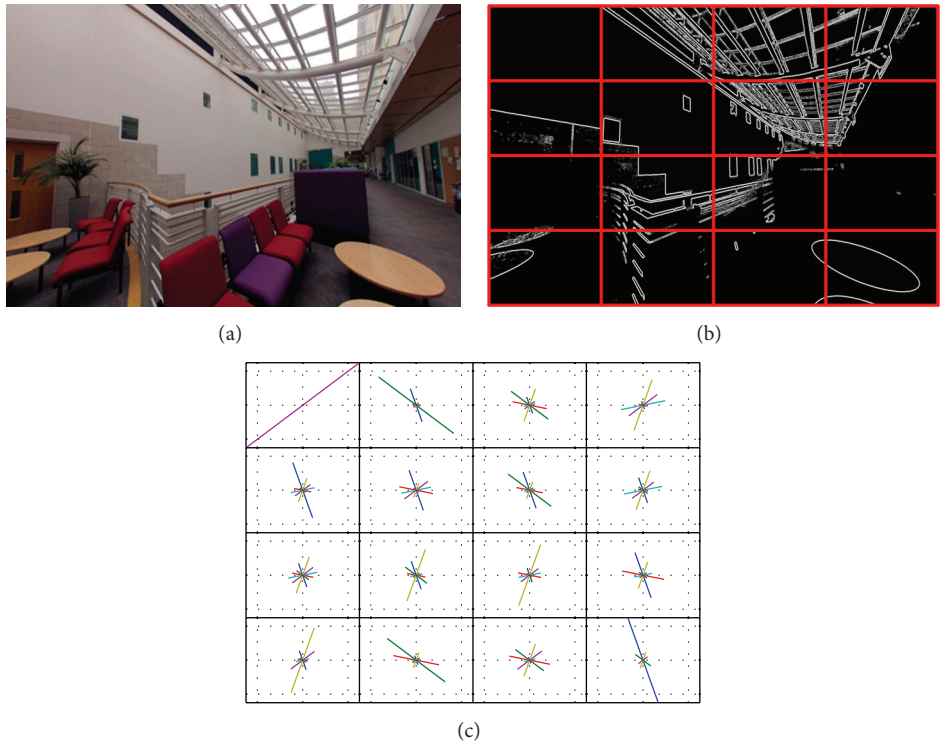


Figure 4: Example of HOG description process. (a) Original image, (b) gradient of the image and cell division, and (c) histogram of oriented gradients of each cell.

finding the correct position using recall and precision measurements [40] for the three nearest neighbours. The Fourier signature is almost invariant to the image resolution, whereas HOG presents a notable reduction of the precision using the smallest resolutions.

The main criteria for the selection of the descriptor is the precision in image association. For that reason, HOG becomes the less appropriate descriptor, although its memory and time performance are favorable. The Fourier signature presents a high precision in the position estimation for all image sizes, but it is lower than Gist-Gabor. Moreover, the size of the database created when we use the Fourier signature is higher than the other techniques. For those reasons, the descriptor selected is Gist-Gabor. The results obtained using the fifth image size (64×32 pixels) show an appropriate compromise between time and memory requirements and precision.

4. Multiscale Analysis

This section describes the multiscale analysis. During the matching process between the images of the nodes and the routes, the nodes might be too separated for a correct association, especially in the route locations that are halfway between nodes. As a consequence, the appearance of the route images could present insufficient similarity with the nodes scenes to find a reliable retrieval in the node database. The aim of the multiscale analysis is to improve the association accuracy and to estimate the relative position between images making use of the global appearance descriptors.

Given two images, this technique carries out the comparison of several zooms-in of the central area of each image at different scales. Figure 10 shows the field of view of a camera when it moves forward perpendicularly to its projection plane. We can appreciate that the scene in the ahead position, represented in blue, corresponds to the central area of the field of view associated with the first position, represented in orange. If we select the central area of the orange image and rescale it to the original image size (simulating a digital zoom), the appearance regarding the second image (the blue) increases. Figure 11 illustrates an example of this idea. It includes two images captured during the forward navigation movement of robot following a route (Figures 11(a) and 11(b)). In the figure is also included a zoom-in of the central area of (a) (Figure 11(a′)). We can appreciate that the zoomed image (a′) is more similar in appearance to (b) than the original scene (a).

The similarity between scenes is measured using global-appearance descriptors (Section 3). After the comparisons with different scales, we select the association with the lowest image distance (i.e., the nearest neighbour), since it denotes the most similar images using the global appearance.

The scales of the two images matched during the association process provide information about their relative position. Specifically, the algorithm uses the difference of the scales to estimate the distance between images.

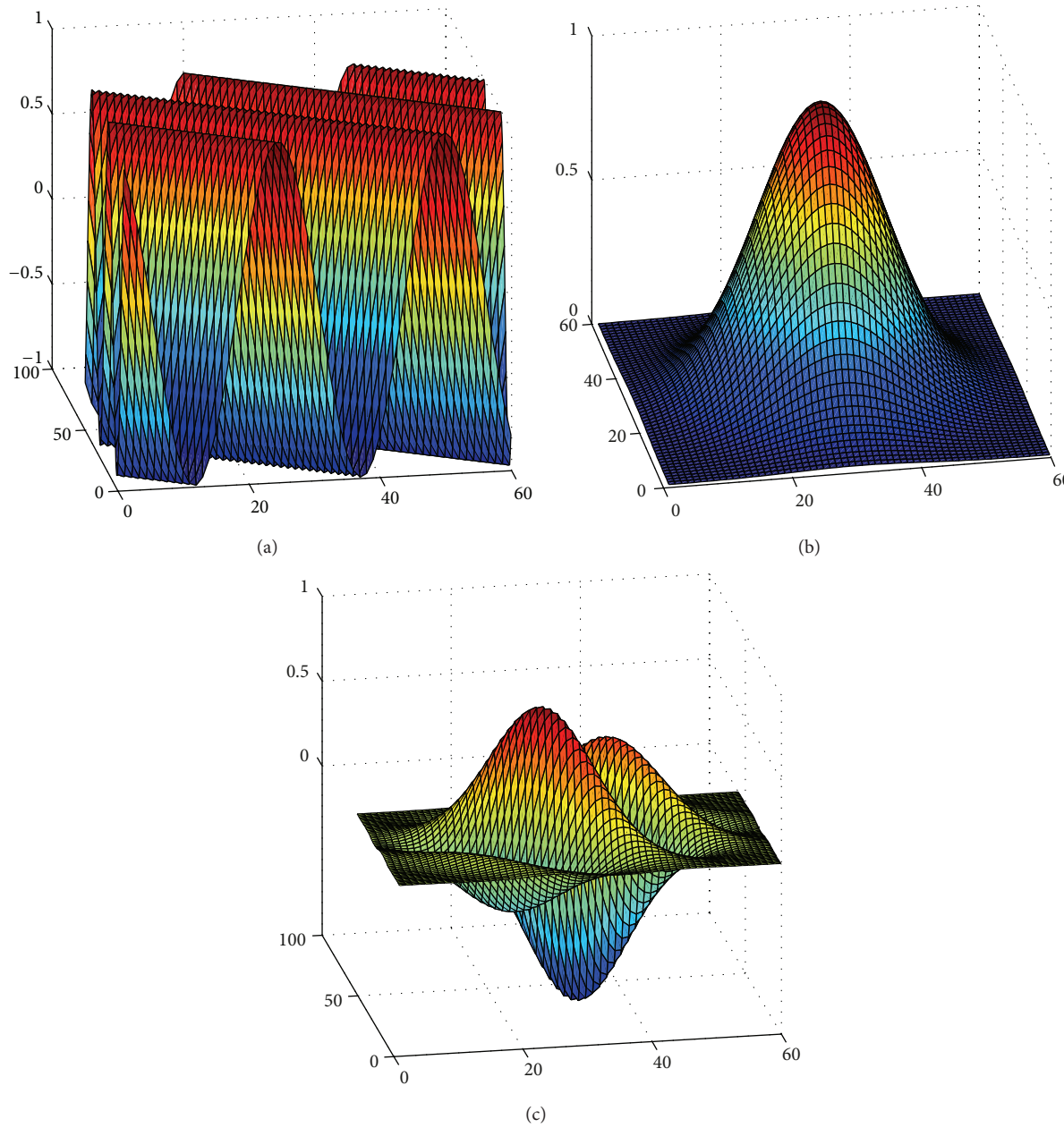


Figure 5: (a) Complex sinusoid, (b) Gaussian envelope, and (c) Gabor filter resulting of the convolution of both functions.

Figure 6: Division of the filtered image in cells and estimation of the mean intensity value of the pixels within each cell.

Figure 12 illustrates this process. The example includes four images of a route captured sequentially as the camera moves forward. In the example, we aim to estimate the topological distance of the four scenes regarding Scene 1, which is our reference image. For that purpose, we estimate different scales of Scene 1, compute their global-appearance descriptors, and compare them with (a) Scene 1, (b) Scene 2, (c) Scene 3, and (d) Scene 4 without zoom.

Since these images are captured sequentially, each scene is geometrically more separated from Scene 1 in the real world.

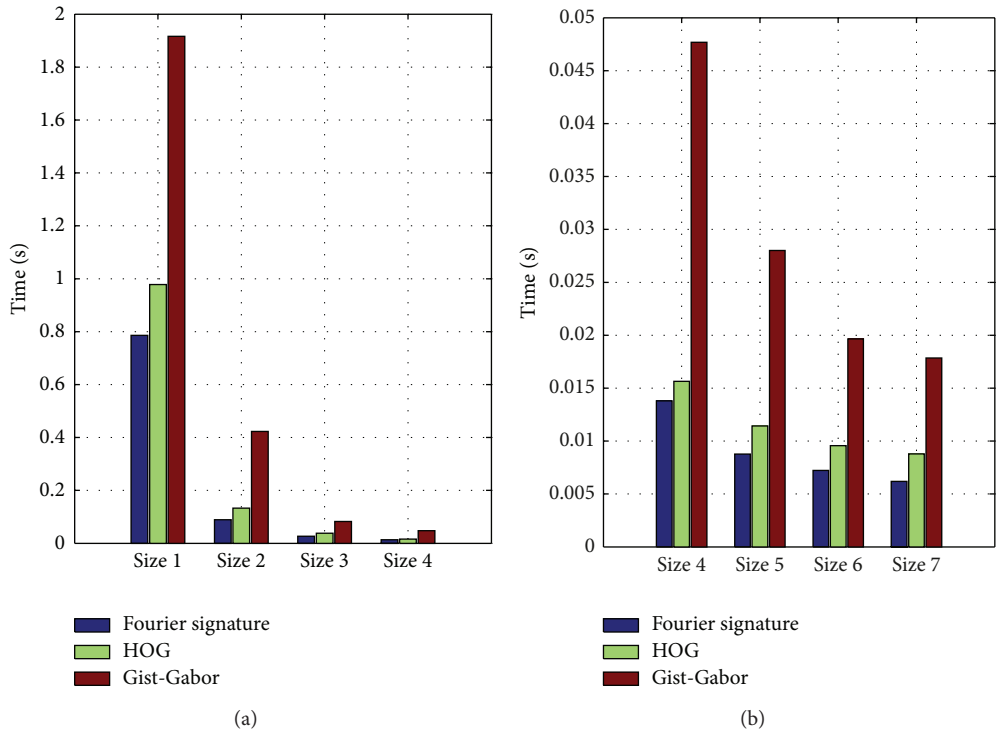


Figure 7: Necessary time to build each descriptor depending on the image size.

In other words, Scene 3 is more separated from Scene 1 than Scene 2, and Scene 4 has been captured in the most distant point from Scene 1.

On the other hand, Figure 12 includes on the right side a graph that represents the image distances of the four scenes versus the different scales of Scene 1.

The scale factor (s) is the quotient between the original resolution of the image and the size of the area we select. For instance, if the resolution of the image is 32×64, a scale equal to $s = 2$ is supposed to select the 16×32 central pixels.

In the graph, we highlight the scale of Scene 1 that presents the minimum image distance for each scene of the example. Note that lower image distances (i.e., Euclidean distance between descriptor) denote higher similarity between scenes.

As expected, the minimum image distance comparing with Scene 1, which is the same image than the reference scene, is obtained using a scale equal to 1, that is, when no zoom is applied. Regarding the comparisons with the other scenes, we can realize that the minimum image distance is obtained for higher zoom scales as the scene is more separated geometrically in the real world from the reference image (Scene 1).

Therefore, there is a direct correlation between the scale where the minimum image distance is obtained and the geometrical distance of the scenes in the real world. In our map building and localization algorithms, we use the difference of scales as topological distance between scenes.

Moreover, as seen in Figure 12, we obtain also a reduction of the image distance when comparing two images, which means that we increase the similarity of the compared scenes using the global appearance.

This increase in the similarity between images turns into an improvement of the precision in the image association task. The map building and navigation algorithms proposed in the following sections of this work rely on the matching between the images of isolated positions in the environment (the nodes) and images acquired along routes. For that reason, an improvement of the association precision is important.

To measure this improvement, we study the association between 352 node images and 172 images of routes. We compare each image of the route with all the node images and select the association with minimum image distance (the nearest neighbour). We consider that the association has been correct when the selected node is the nearest in geometric distance in the real world. Figure 13 shows the recall-precision results using the multiscale analysis and without it. Thanks to the introduction of the multiscale analysis, the precision in correct node retrieval increases 14%.

Therefore, the multiscale analysis improves the association between images and provides a measurement of topological displacement between two images by means of the zoom scales (s).

5. Map Building

This section details our topological map building algorithm. It starts with a database that contains the descriptors of all the images of the nodes, with no information of their spatial

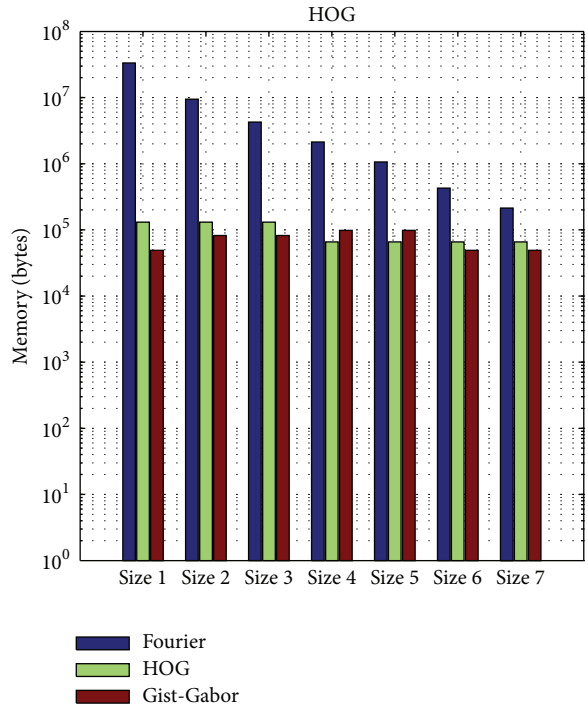


Figure 8: Database memory requirements using several image sizes and descriptors.

distribution. In the database, the images of the same node are stored consecutively, but the order of the nodes does not provide information about their spatial layout.

We also have different routes of images captured while the robot navigates along the nodes. The scenes of the routes are ordered as they are captured during the navigation, and the algorithm incorporates their information sequentially, that is, in the same order they are captured.

The aim of this algorithm is to select the nodes of the database as they are associated with the images of the routes using the global appearance descriptors, to establish the adjacency relationship between nodes, to define its orientation, and to estimate the distances between nodes using the multiscale analysis.

After the map building process, we obtain a graph that represents the spatial distribution of the nodes, and the edges are the adjacency relations between those nodes.

5.1. Estimation of the Relative Position between Nodes and Route Images Using the Multiscale Analysis. During the map building, the algorithm uses both the multiscale analysis to compare each image of the route with the images of the nodes. Given a route image, the matching process carries out the comparison of different scales of that image with several scales of the node images. After the comparisons, we select the experiment with the minimum image distance. s_n and s_r represent the specific scale factors of the node and the route images, respectively, obtained with the multiscale analysis.

These two scales permit determining the relative position between the image of the node and the route. The topological distance (l) between the route image and the node can be estimated as

$$l = s_n - s_r. \tag{9}$$

Following the example included in Figure 14, when the route image is in front of the node (example Node 1), the comparison with the highest similarity between scenes is obtained doing a zoom-in of the node image (Figure 14(a′)). Hence, $s_n > s_r$, obtaining a negative topological distance ($l > 0$). On the contrary, in the image of the route is situated backwards the node (example Node 2), the minimum image distance is obtained when we compare the image of the route using zoom-in with the image of the node without any zoom. In that case, according to (9), we obtain $l < 0$. Therefore, the topological distance l not only provides information about the relative distance between the nodes and the route images but also the direction of their distance by means of its sign.

In Figure 15, an example of a reduced experiment of node retrieval and distance estimation is shown. It includes two node images and nine images of a route whose path coincides with the nodes position. The results show the nearest node n, the scales of the node image (s^n) and the route image (s^r) estimated using the multiscale analysis, and the topological distance l. In this example, we have omitted the estimation of the orientation, since the route follows a straight line. In the localization results, we can see that the topological distance l is negative when the route images are backwards the nearest node and positive when they are ahead the node.

5.2. Association between Routes and Nodes Images. The first step in the map building algorithm is the matching between

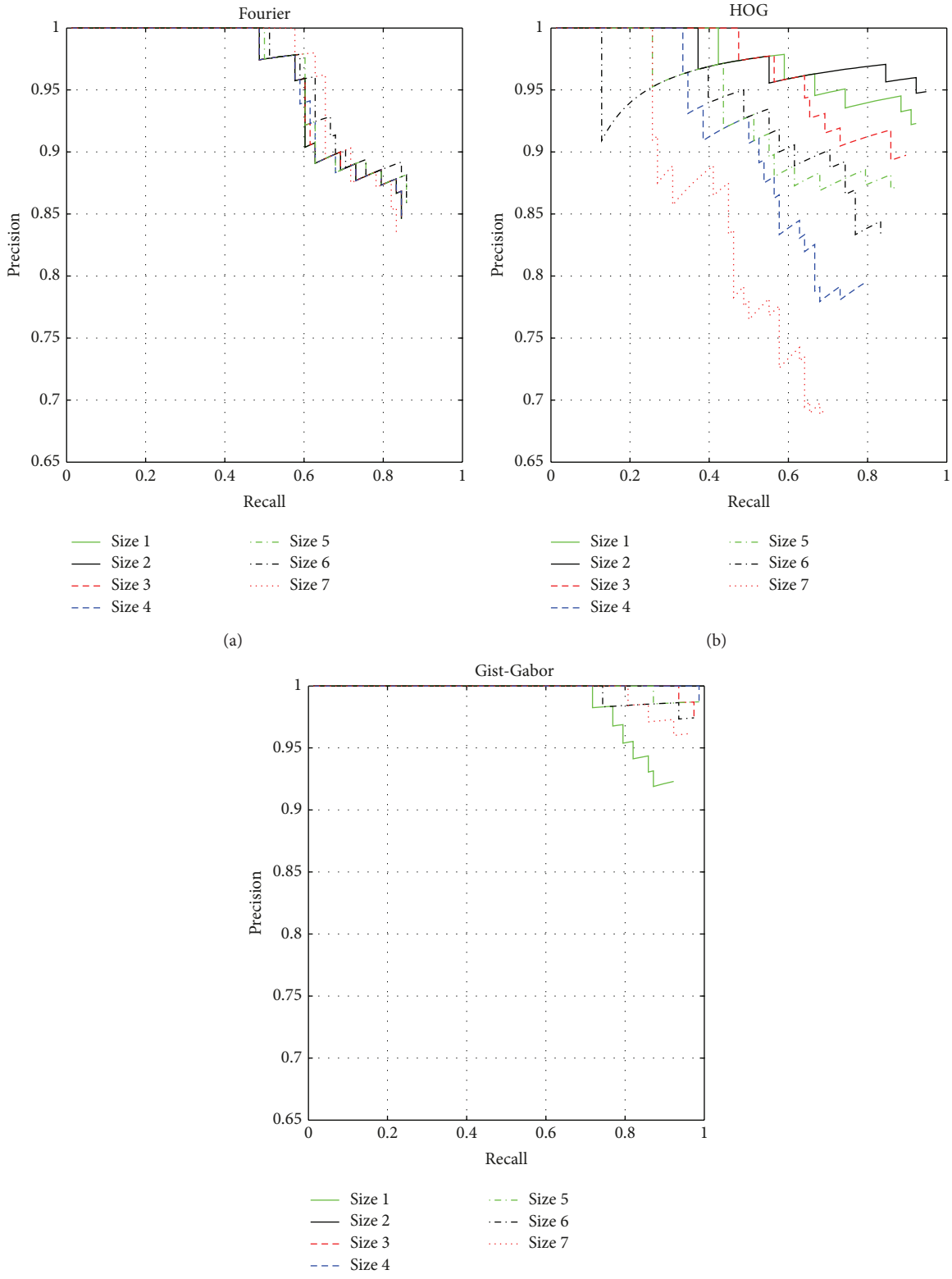


Figure 9: Recall-precision graphs in location retrieval considering the three nearest neighbours for different image's sizes using (a) Fourier signature, (b) HOG, and (c) Gist-Gabor.

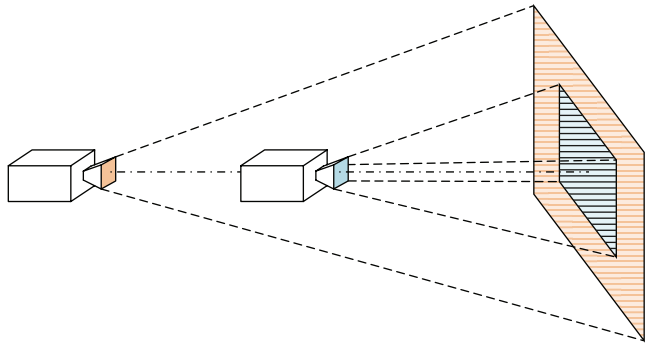


Figure 10: Representation of the field of view of a camera considering a movement perpendicular to the projection plane.

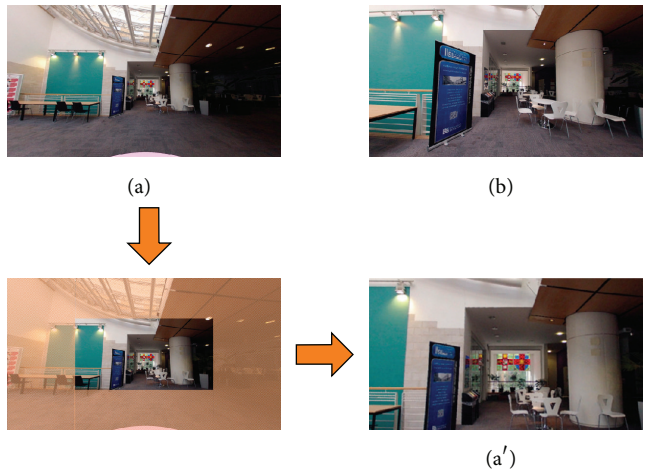


Figure 11: (a) Image captured in a route, (b) image captured in front of image (a), and (a′) zoom-in of image (a).

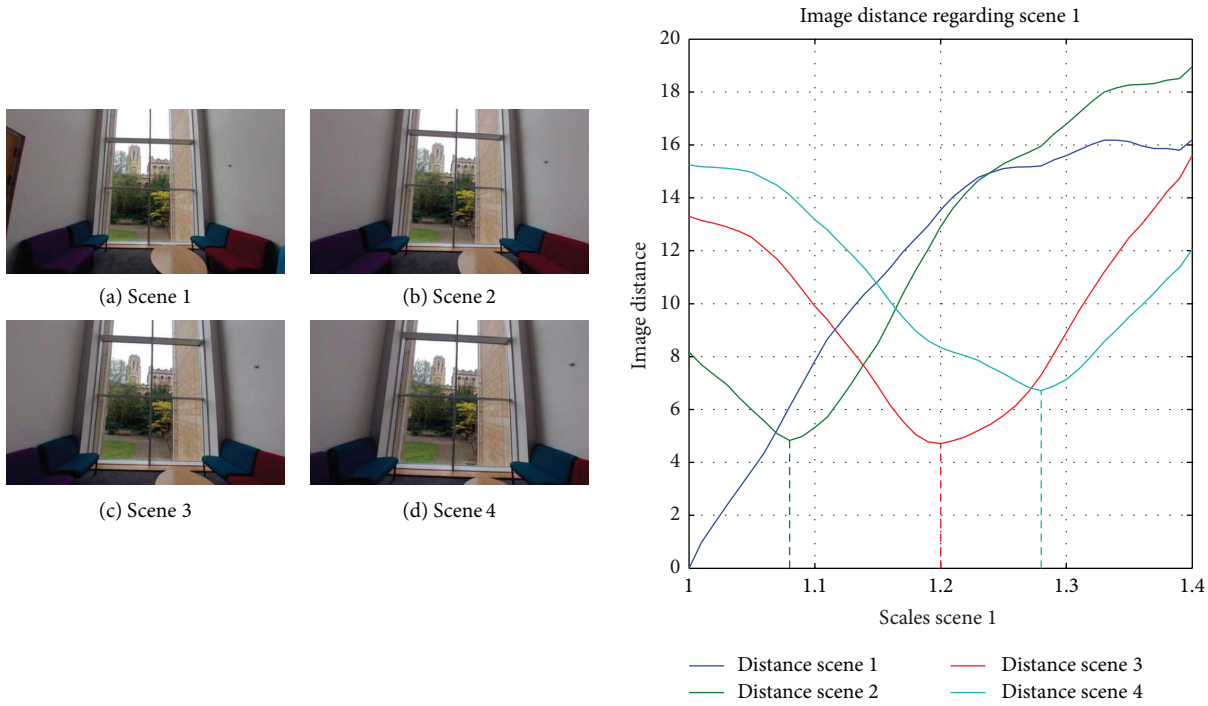


Figure 12: Consecutive scenes of a route of images and image distance of the scenes regarding different scales of Scene 1.

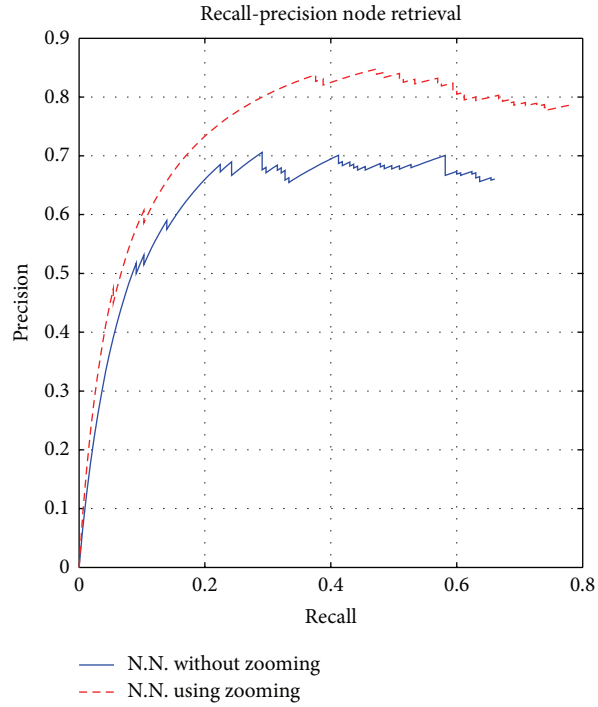


Figure 13: Recall-precision of the nearest neighbour in node retrieval using Gist-Gabor as a descriptor. Comparison with and without the multiscale analysis.

the routes' and the nodes' images. This association is used to decide whether a new node is included in the map and is based on the image distance using the global appearance of the scene.

The algorithm can be summarized as follows.

(i) First, we create the map retrieval database. For that purpose, the algorithm computes the descriptors $z^n \in \Re^{1xy}$ of the node imagery (including different scales of every image). y denotes the number of components of each descriptor.

(ii) The descriptors are stored in columns of a matrix, which represents the map database, $\mathbf{Z} = [z_1^n, z_2^n, \ldots z_i^n, \ldots, z_m^n]$, and m is the number of images included in the database, that corresponds with the product of the number of nodes, orientations per node, and number of zoom scales per image.

(iii) Since the descriptors are stored following the same order as the database images, it is possible to know the node n, orientation in the node θ, and zoom factor scale s^n of each descriptor included in the database, since they are function of the position of the descriptor in the matrix $\mathbf{Z}$. Denoting i as the number of column in $\mathbf{Z}$,

$$[n, \theta, s_n] = f(i). \tag{10}$$

It should be noted that the order in which the nodes are stored in the database does not provide information about its spatial distribution. The position of the nodes is totally unknown to the system when the algorithm starts.

(iv) When a new route image arrives, the algorithm computes its descriptor z^r, and it calculates its Euclidean distance d with all the descriptors included in $\mathbf{Z}$:

$$d_i^r = \sqrt{\sum_{a=1}^{y} \left(z_{i,a}^n - z_a^r\right)^2}, \quad i = 1, \ldots, m. \tag{11}$$

(v) The image distance d_i^r is used as a classifier. The algorithm selects the nearest neighbour and is associated with the minimum distance d the corresponding values of n, θ, and s_n.

(vi) The algorithm repeats this process using different scales of the route image (s_r).

(vii) Once we have estimated the image association for the different scales s^r, we order the results regarding d and select the experiment with the minimum distance.

(viii) Finally, we save the parameters corresponding to the minimum image distance (d). From every route image, we obtain the information vector:

$$[n \;\; d \;\; \theta \;\; s_n \;\; s_r]. \tag{12}$$

5.3. Graph Creation. The process of including a new node in the map starts with the information vector described in (12). We obtain a vector from every route image, and as we study a new image, we add the new information vector to an array. The decision of including a node in the map involves the last 5 route images.

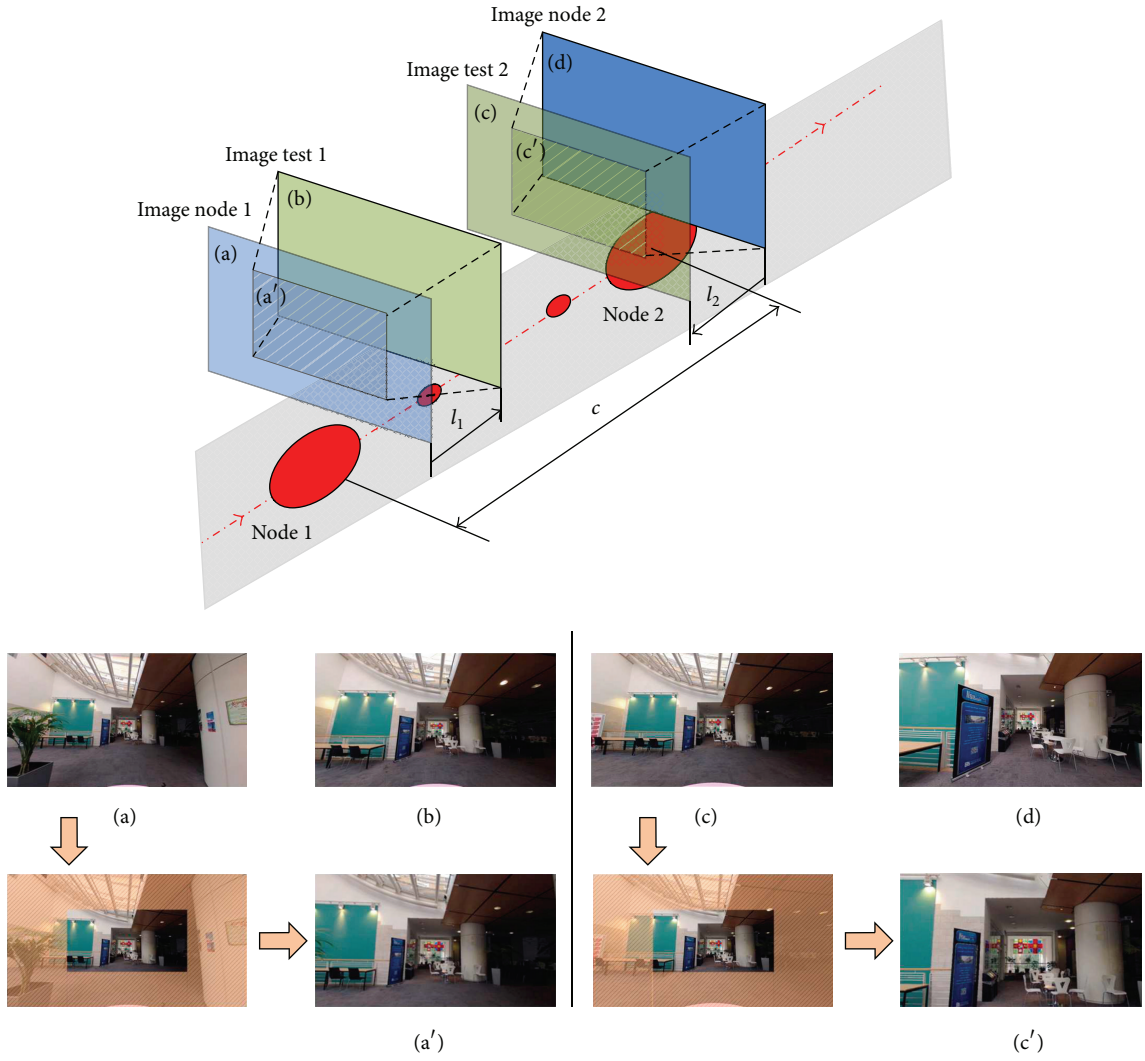


Figure 14: (a) Node image, (a′) zoom-in of the node image (a), and (b) route image located infront of the node image (a). (c) Route image, (c′) zoom-in of the route image (c), and (d) nearest node image of (c) located in front of that image. l_1 and l_2 are topological distances between a route image and the nearest node, and c is the topological distance between nodes.

M is the number of repetitions of the mode value n_m of the nodes number (n) included in the last 5 node's retrieval, and μ and σ are the mean and standard deviation of all the d included in the information vectors so far. The node n_m is included in the graph if any of these two conditions is achieved:

(i) $M \leq 3$,

(ii) $M = 2$ and $d_{n_m} < \mu - \sigma$.

Therefore, the algorithm includes a new node if it associates the same node in 3 of the 5 last route images or in 2 of them but with a los image distance (what denotes a highly reliable association).

When the image association has an image distance value $d > \mu + 2\sigma$, the information vector is not taken into account, since a high value of d indicates low reliability in the association.

To know the connections between nodes, we create the adjacency matrix $A \in \Re^{N \times N}$, being N the number of nodes. A is a sparse matrix with rows labelled by graph nodes, with 1 denoting adjacent nodes, or 0 on the contrary. Supposing we have included the node n_1 in the graph and the next node found is n_2, $A_{n_1,n_2} = 1$.

Regarding the topological distance between the nodes in the graph, we make use of the image scale factors. To estimate the distance between two consecutive nodes, the algorithm uses the following information: the topological distance of the last route image in which the first node is detected (l^l), and the scale difference between the first image of the route matched with the next node (l^f). It is worthy of recalling that, as stated above, since the last route scene where a node is detected is due to be in front of that node, the value of l^l will be positive. On the contrary, as the first route image where a new node is matched is usually behind the node, l^f is expected to be negative. So then, the topological distance $c_{n_i,n_{i+1}}$ between

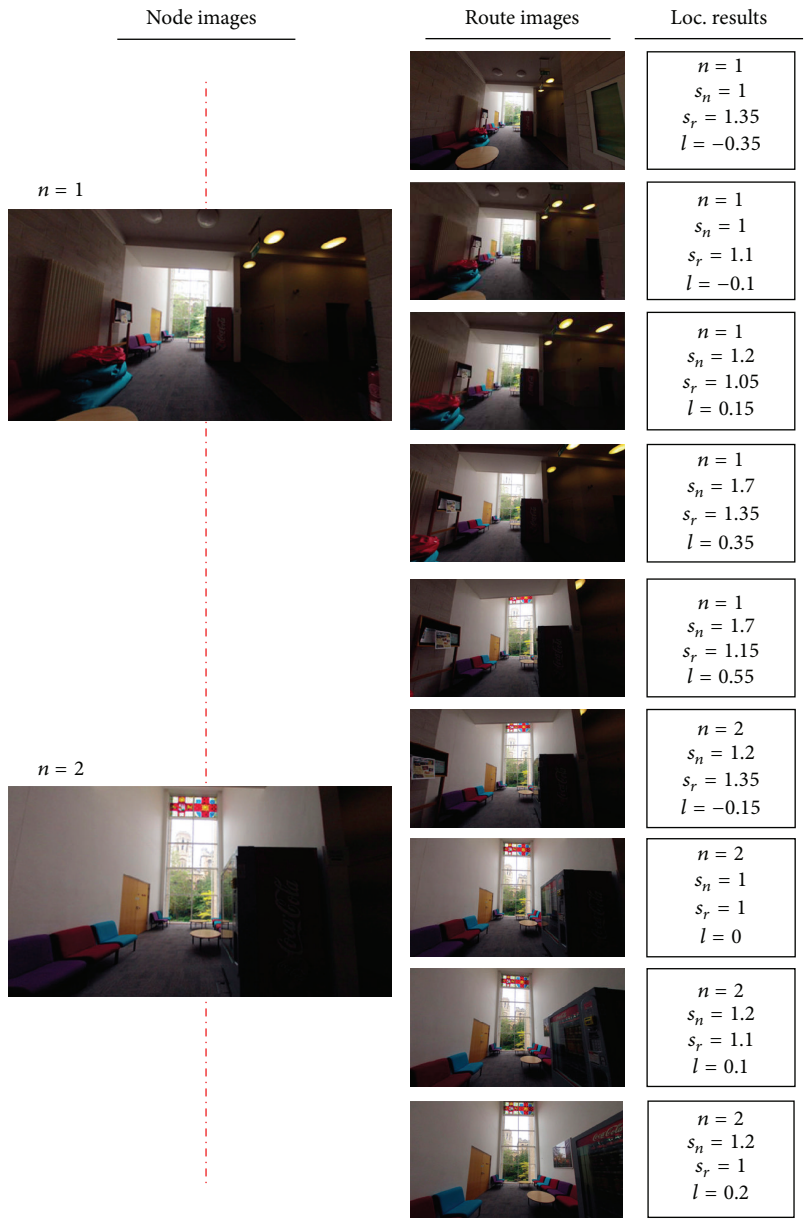


FIGURE 15: Example of image retrieval experiments carried out using two route images and nine scenes of the route that connects both nodes. In the right side, the localization results are shown, including the nearest image (n), the node scene scale factor (s_n), the route scene scale factor (s_r), and the relative topological distance between the node and the route images (l).

a node n_i and n_{i+1} takes into account the sign of the distances regarding the relative position of the route images and the nodes, and it is defined as

$$c_{n_i,n_{i+1}} = l^l_{n_i} - l^f_{n_{i+1}}. \tag{13}$$

Following the example included in Figure 15, l^l_1 corresponds with the topological distance obtained in the fifth route image (the last one where the node 1 is detected) and l^f_2 with the topological distance of the sixth scene (the first where the node 2 is retrieved). Then, $c_{1,2} = 0.55 - (-0.15) = 0.7$.

To build the graph, it is necessary to incorporate information about the orientation. We suppose that the routes follow a straight path until we detect a change of direction in one of the nodes. $\theta^f_{n_i}$ denotes the orientation associated with the first route image where the node i is retrieved, and the output angle $\theta^l_{n_i}$ is the direction of the last image where the same node is detected. The difference of these angles provides a phase lag that coincides with the change in the direction of the graph:

$$\Delta\theta_{n_i} = \theta^l_{n_i} - \theta^f_{n_i}. \tag{14}$$

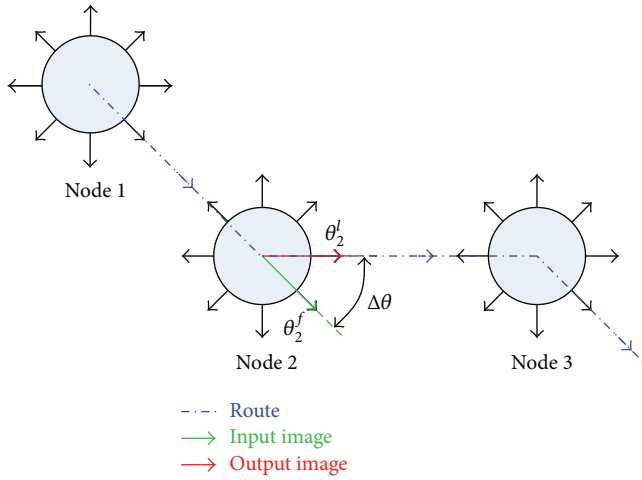


FIGURE 16: Phase change estimation in a node. θ_2^f is the direction of the first image retrieved of node 2 and θ_2^l is the direction of the last image retrieved of node 2. $\Delta\theta$ is the phase lag.

Moreover, $\theta_{n_i}^l$ is the direction the robot has to follow in order to arrive from node n_i to the node n_{i+1}. Figure 16 illustrates the phase lag between two nodes.

We set the orientation of the map by defining the direction of the output image in the first node. That direction determines the global orientation system of the map. The orientation of the graph is updated in every node with the phase lag defined in (14). Since we have a global direction orientation system defined, we can compute for each node the difference of orientation between its local system and the global. For instance, if the input direction of a node is 0° in the global system, and it corresponds to 90° regarding the local system, we have a phase lag of 90° for that node. That way, we can include the new nodes and orientate them according to our global reference system.

When a new route is studied, the algorithm initializes a new coordinates system for its nodes. That route will be analysed independently of the global graph until a common node is found. Using the position and orientation of the common node regarding both systems, we are able to add the new nodes of the current route to the global graph. If two routes share a common path and we match nodes of the map database that have been previously included in the global map, the topological distances $c_{n_i,n_{i+1}}$ between those common nodes are estimated again, and the results are taken into account in the graph by calculating the mean of the new estimation, $c_{n_i,n_{i+1}}$, with the previous estimations. The mean will be weighted by the number of times that the same distance has appeared.

Therefore, our map building algorithm takes advantage of the information provided by the routes in order to obtain the relative position of the nodes by matching the sequence of images with the nodes descriptors database. It uses that information to estimate the adjacency relations, but it also gives information about the relative distance and position between them using the multiscale analysis.

6. Path Estimation Algorithm

Once we have carried out the map building and we obtain the graph that represents the layout of the nodes, our aim is to estimate the path of the routes that the robot follows during the navigation in this graph. We can divide the localization of the robot in the map in two main steps: first, we carry out a coarse estimation, identifying the nearest node and the orientation. Note that the orientation of the robot is determined using the phase of the node image associated with the route image. The algorithm uses a weighting function to penalize associations that are geometrically far from the previous route pose, since consecutive route images should be located nearby in the graph.

If the localization of the route images is based only on the matching with the nodes in an image retrieval process, the localization accuracy will be limited to the node positions. In order to obtain a more accurate estimation of the pose, the second step in the localization algorithm includes the multiscale analysis. Specifically, we apply this technique using the current route image and the associated node image. That way, the algorithm is able to find the relative position between both images and to extend the possible position values to intermediate position of the nodes locations.

When a route image arrives, we compare several zoom scales of this image with the nodes database, $\mathbf{Z}$, that includes the descriptor of different scales of the nodes images. The association between the route image and the database is determined again using the nearest neighbour regarding the image distance (d). Since the test images come from a route path, we can suppose that the distance and phase lag between consecutive images should be not high. For that reason, in order to improve the localization of the route images, the algorithm introduces a weighting function in order to penalize the probability of finding the current location or orientation far away from the previous image pose.

6.1. Weighting Function. As stated at the beginning of this section, we introduce a weighting function in the algorithm to improve the localization accuracy of the route images in the topological map. This function reduces the probability of finding the location of the current nearest node distant from the previous image pose. Since the image association criteria are the nearest neighbour, the weighting function increases the image distance of the associations whose node image pose is distant from the last robot pose. In this way, we reduce the likelihood of selecting them as the current nearest node.

The weighting function is composed of 2 terms: the first one takes into account the topological distance between consecutive route images in the graph and the second their phase lag.

The first term uses the positions of the nearest nodes associated with the previous and the current route images in order to estimate their topological distance in the map. The adjacency matrix A allows us to find out the shortest path between two nodes in the map. Since we have a connected graph, we can always find a path that connects any 2 nodes of the map. c_{n_1,n_2} represents the cost of traversing 2 adjacent nodes $n_1, n_2 \in A$ (that corresponds with the topological distance between nodes, defined in (13)) and $P_{n_i,n_j} = [n_i, \ldots, n_j]$ the sequence of nodes of the shortest path that connects n_i and n_j, the cost C_{n_i,n_j} associated with the sequence of nodes P_{n_i,n_j} can be defined as

$$C_{n_i,n_j} = \sum_{n_i}^{n_j} c_{n_i,n_{i+1}}. \tag{15}$$

The second term takes into account the phase lag between consecutive route poses.

Finally, the weighting function between two images can be defined as

$$w\left(n_i, n_j, \theta_{n_i}, \theta_{n_j}\right) = w_1 \cdot C_{n_i,n_j} + w_2 \cdot \left|\theta_{n_j} - \theta_{n_i}\right|, \tag{16}$$

where w_1 and w_2 are constants that module the weight action of the topological distance and the phase lag, respectively.

As (16) shows, the weighting value between 2 images depends on the cost to traverse the path that connects their respective closest nodes and their orientation difference.

6.2. Route Images Localization in the Graph. First, the algorithm computes the image distance between the current route image and the nodes images using (11). From it, we obtain d_i^r, $i = 1, \ldots, m$, that represents the image distance of the route image with every image included in the map database, $\mathbf{Z}$. It includes the descriptors of all the nodes images with different scales.

Since the map database includes different zoom scales of image of the nodes, each descriptor included in $\mathbf{Z}$ has a value of n, θ, and s_n associated (10). The algorithm compares the current route image descriptor with $\mathbf{Z}$, classifies the results regarding the value of image distance d, and selects the k-nearest neighbours. Then, this process is repeated using different zoom scales of the route image s_r.

After that, we update the image distance values d of the k neighbours selected using each scale s_r using the weighting function:

$$d' = d \times w\left(n_i, n_{i-1}, \theta_{n_i}, \theta_{n_{i-1}}\right), \tag{17}$$

with n_{i-1} and $\theta_{n_{i-1}}$ the nearest node and orientation of the previous route image and n_i and θ_i the nearest node and orientation of each neighbour selected in the matchings. The weighting value may change for every neighbour, since n_i and θ_i might be different in every particular case.

When all the image distances have been updated, we classify again the results regarding d' and choose the Nearest Neighbour. With the information associated with the retrieval, we find out the closest node n of the route image, its orientation θ, and the scale factors of both the node and the route images (s^n and s^r).

With this data, we can determine the current robot pose in the map. The position is estimated using the nearest node, and the relative position between the matched images, provided by the multiscale analysis and the difference of the scale factors s_n and s_r defined in (9). The direction of advance is provided by θ. Note that θ is the orientation of the node image regarding the local reference system of each node that must be corrected with the phase lag between the map global system and the node reference system, estimated previously during the map building process.

7. Experiments and Results

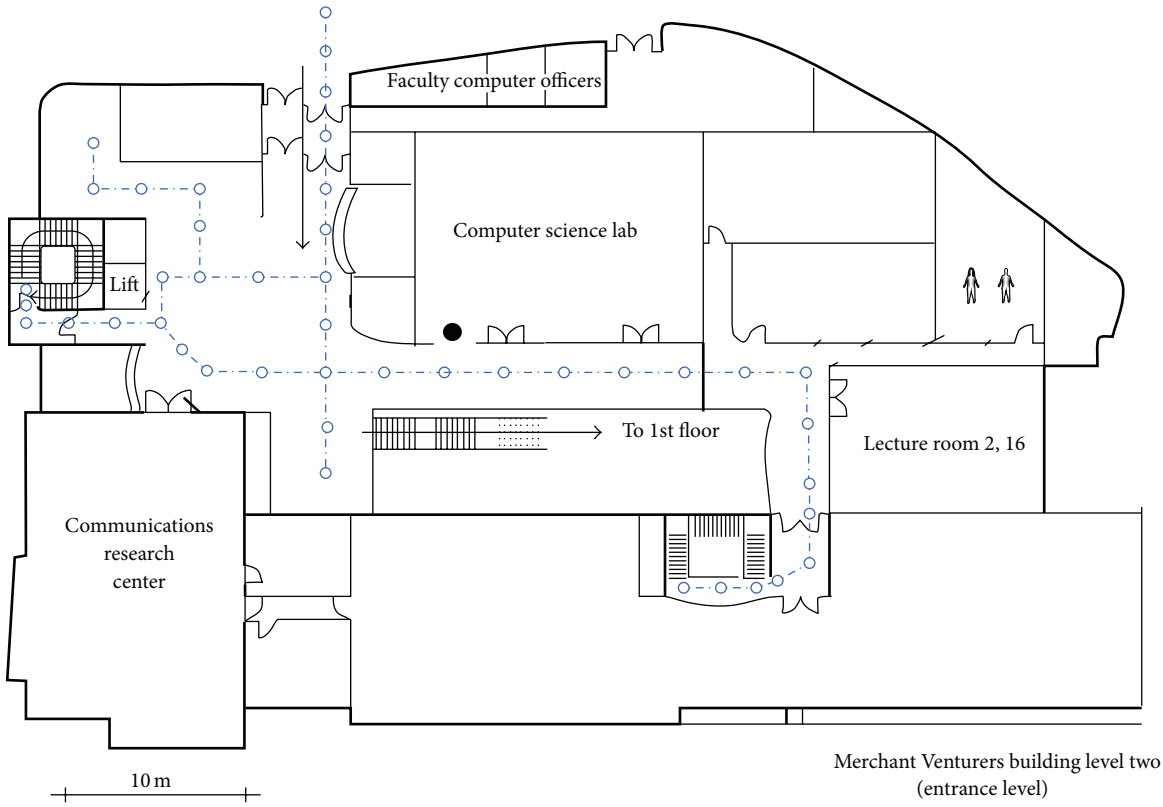
This section details the database used during the experiments and the results of the map building and route path estimation using the multiscale analysis and the global appearance of images. As stated at the end of Section 3, the technique selected to describe the global appearance of the images is Gist-Gabor, and the image resolution is 32 × 64 pixels.

7.1. Dataset: Nodes and Routes Images. Two different databases have been captured. They correspond to common areas of the Merchant Venturers Building of the University of Bristol.

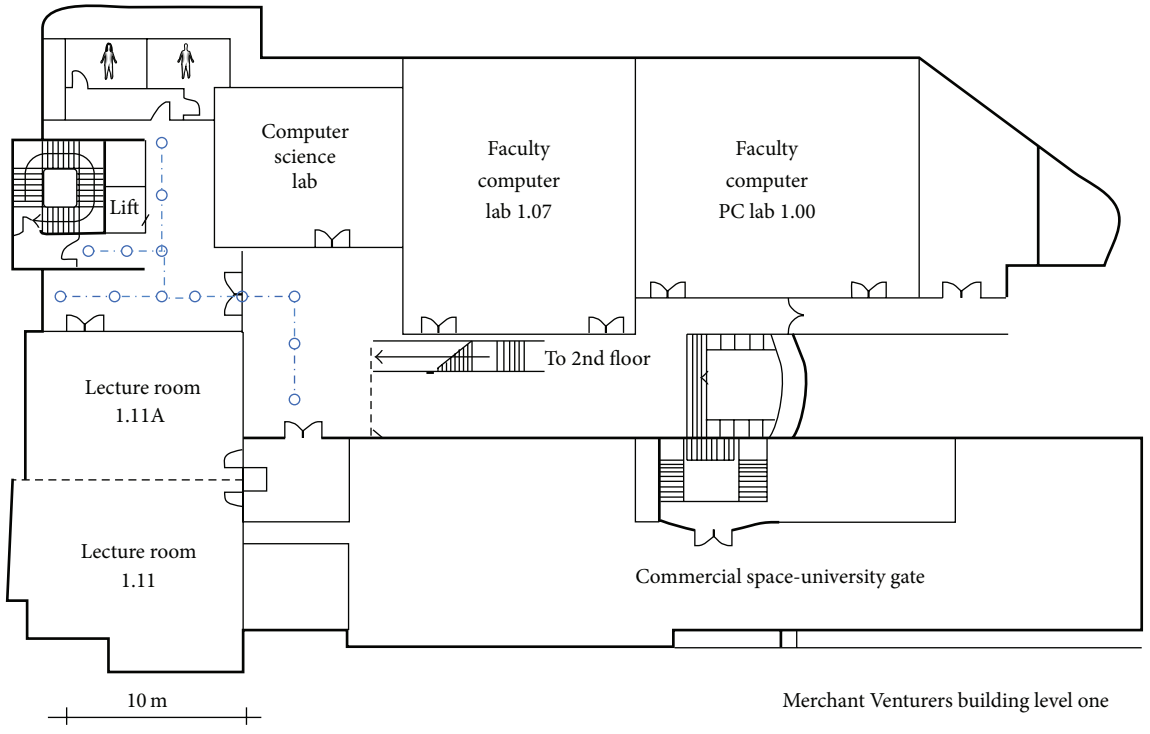
Each database is composed of a set of nodes, and different routes of images are distributed along the areas where the nodes were captured. Note that each node has 8 images, with a phase lag of 45 between consecutive images, covering the complete file of view around the position of the node.

The experiments are divided into two different areas. The number of images of each area appears in Table 2, and the real distribution of the nodes in Figure 17. The actual distance between consecutive nodes is 2 meters as a rule, but in places where an important change of appearance is produced, that is, changes of direction or crossing a door, a new node is captured independently of the distance with the previous node. For that reason, the distance may be lower.

Regarding the routes, the frequency of image acquisition is higher in the routes of Area 2. The images are taken every 0.5 meter in Area 1 and every 0.2 meter in Area 2. We increase the capture rate at turnings. We take a minimum of four images per position when a change of orientation is produced.



(a) Area 1



(b) Area 2

Figure 17: Representation of the graphs in the plane of each navigation area.

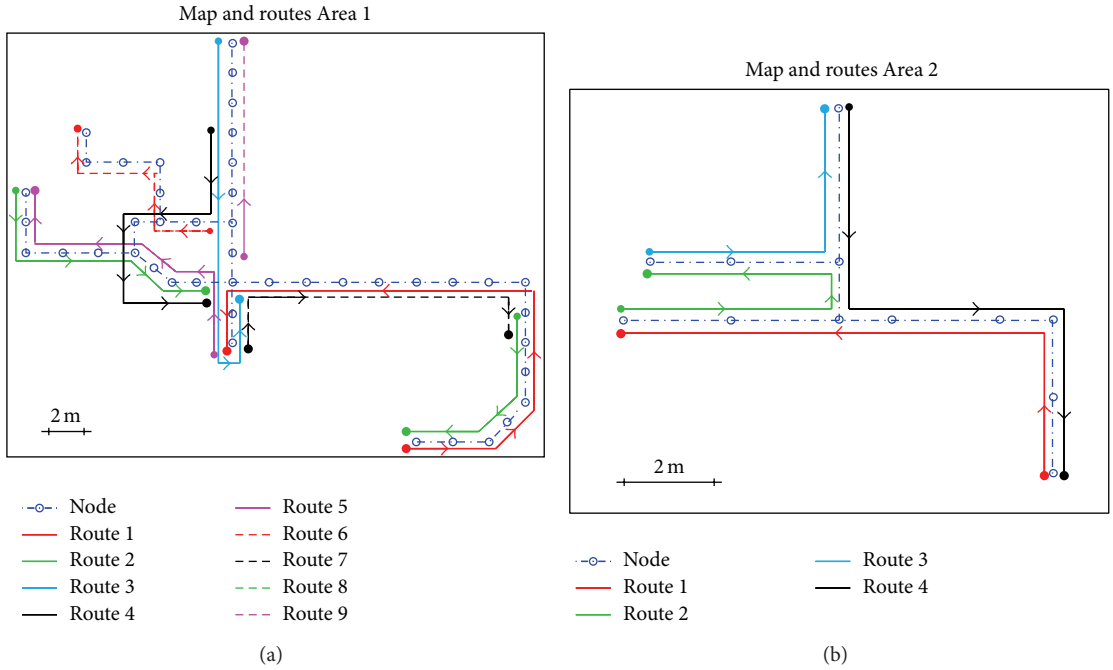


Figure 18: Synthetic distribution of the nodes and routes for (a) Area 1 and (b) Area 2.

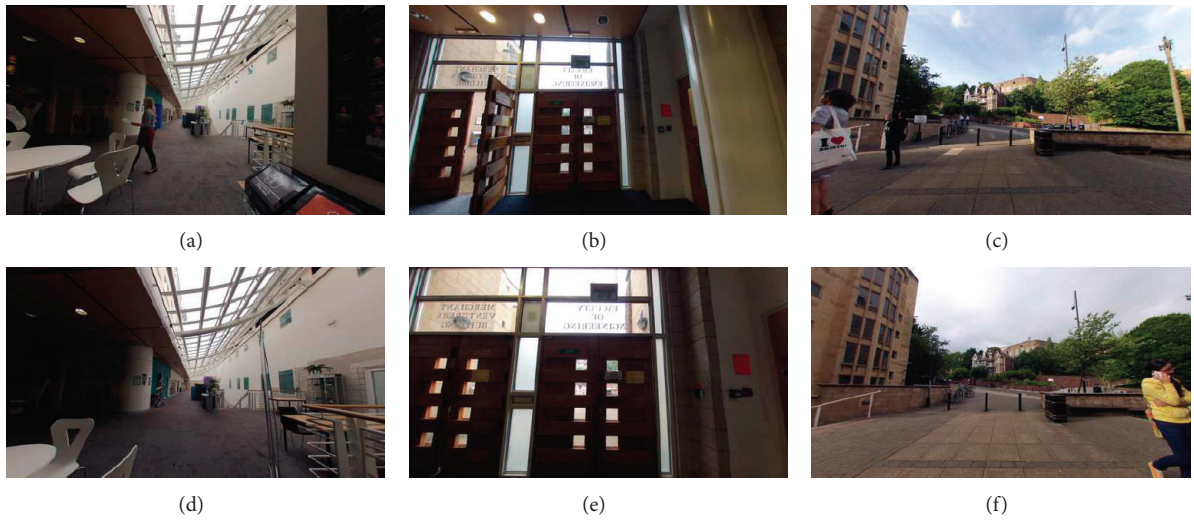


Figure 19: Examples of images used in the experiments: (a), (b), and (c) are routes images and (d), (e), and (f) are the corresponding nearest nodes images.

In Area 2, this frequency increases with a minimum of 6 images per position. Figure 18 shows the distribution of the nodes and the routes in a synthetic representation. Figure 19 presents some examples of node and route images. They show typical situations of real applications, such as changes in illumination conditions and movements of the furniture and occlusions produced by people moving in the area. The system must be able to cope with these situations.

7.2. Map Building Results. Figure 20 presents the nodes graph of (a) Area 1 and (b) Area 2. It has been obtained after running our algorithm. We can appreciate that the algorithm is able to estimate the connections between nodes, with a similar distribution regarding the real layout in the both areas. Area 1 has been the most challenging due to the higher number of nodes, the transition from different rooms, and the loop closure in the map. In the loop closure, the graph representation slightly differs from the real layout. However, although the map loses some accuracy, the navigation of the robot is not affected. The robot can navigate from one node to the other by knowing the node output image that connects the first node to the second one, and this does not depend on the graph layout.

It is important to remark that the algorithm needs a minimum number of route images between nodes. Otherwise, a node might not be included in the map.

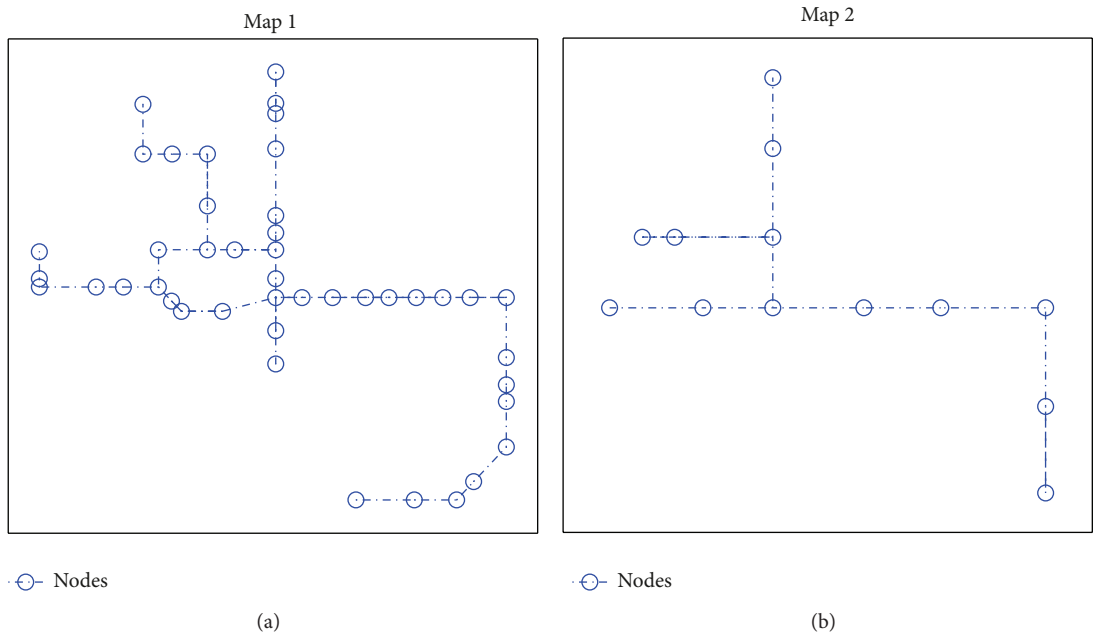


Figure 20: Graph representation of the nodes arrangement obtained with the map building algorithm for (a) Area 1 and (b) Area 2.

Table 2: Number of images per area.

	Number of images (Area 1)	Number of images (Area 2)
Nodes	352	52
Route 1	110	100
Route 2	50	72
Route 3	67	66
Route 4	58	125
Route 5	62	—
Route 6	46	—
Route 7	69	—
Route 8	67	—
Route 9	40	—

Table 3: Procrustes analysis results of the graphs obtained in the map building.

	Area 1	Area 2
μ	0.0372	0.0078

We use the Procrustes analysis [41, 42] in order to measure the error of the graphs obtained in the map building. This analysis studies the geometric error between the real layout of the nodes and the layout obtained with our algorithm. It returns a standardized value of dissimilarity $\mu \in [0, 1]$. The lower μ, the more accurate graph. We show the results in Table 3. In both cases, the geometric error is considerably low.

The system is especially sensitive in the phase lag between nodes, since it is based on the angle estimation of the input and output images of the node. For that reason, it is advisable to raise the frequency of the image acquisition in the nodes where there is a change of direction. In the experiments, the maximum value of s^n is 2.5, with a fixed step of 0.1 between consecutive scale factors. Regarding the routes images, s^r has a maximum value of 1.4, with a step of 0.05. We have chosen a small step in both route and node scales since we have given priority to the performance of the results over the computational requirements. The average time per image in Area 1 is 725 ms and in Area 2 is 680 ms. The difference of time requirements is due to the matching of the routes images with the nodes database, since the number of nodes in Area 1 is bigger. The estimation of the Euclidean distance between the new route image descriptor and the descriptors contained in the database supposes the 45% of the total time in the map building process. Area 1 contains more nodes than Area 2, so that its descriptors database has a higher number of elements, what supposes more time to carrying out the retrieval and, therefore, an increase of the global process time in the map building.

The orientation of the global reference system is determined by defining the direction of the output image of the first node. In the experiments, we choose this direction so that the graph has the same orientation that the layout represented in Figure 18. If we had chosen any other direction, the map shape would have been the same, but rotated. Anyway, the orientation of the global reference system does not affect the localization algorithm.

7.3. Path Estimation Results. In order to find proper values of the weighting constants, we carry out a study of the localization performance regarding the values of w_1 and w_2. Figure 21 shows the precision in the node image retrieval varying both parameters. The dataset of the experiments is composed of the images included in the routes 1 and 5 of the map 1. In the precision measurement, we consider that a retrieval has been successful when it selects the node image that corresponds

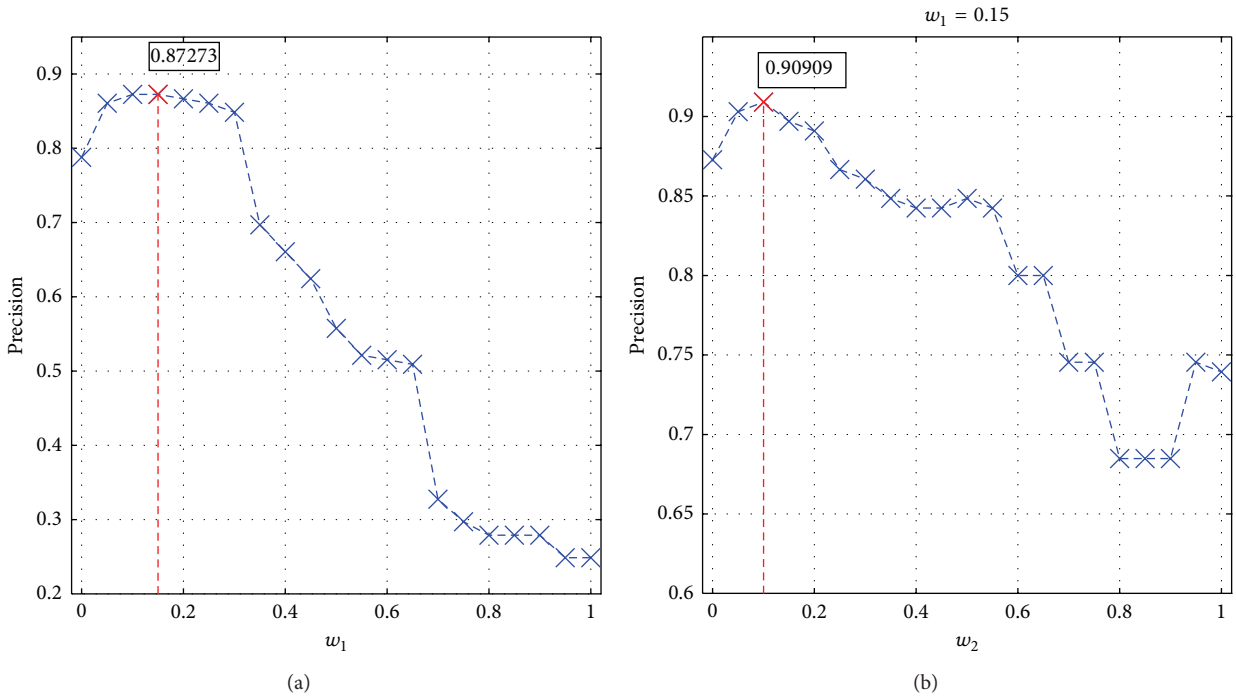


FIGURE 21: Image retrieval precision regarding the image distance weighting parameters. (a) Precision varying the topological distance constant (w_1) and (b) precision varying the phase change constant (w_2).

with both the correct position and orientation. In Figure 21(a) we study the retrieval performance regarding the weighting constant w_1. We can notice an increase of the precision for low values of w_1. Once we have selected w_1, we study the precision varying the parameter w_2. The results are shown in in Figure 21(b). Both graphs prove that the weighting function improves the retrieval precision. However, if we are too restrictive with the position or phase changes, the precision decreases. For that reason, when the constants are given high values, the retrieval accuracy is lower.

In the path localization experiments, the weighting constants are given the values $w_1 = 0.15$ and $w_2 = 0.1$, and we use $k = 10$ nearest neighbours when doing the retrieval of each zoom scale of the route images. The node zoom scale s_n varies from 1 to 2.2 with a step of 0.4. Regarding the route zoom scale, it varies from 1 to 2.2 with a step of 0.3 between consecutive scales.

Figure 22 shows the path estimation of different routes of both areas. The dots in the paths of the routes represent the position of the different images studied. As it can be seen, the algorithm copes with the interpolation of the location in halfway positions between the nodes using the image's scales information. In general, the precision at turnings in the routes decreases. It is also important that, despite the fact that we introduce the weighting function, the algorithm is able to find again the correct position although a previous estimation is not correct, as we can see in Figures 22(a) or 22(c). The result in the path planning of the fourth route of the first area (Figure 22(b)) is also interesting. As we can appreciate in Figure 18(a), the route number 4 presents a variation in its path that differs from the layout of the nodes. However, despite that fact, the path estimation algorithm is able to estimate the position accurately.

Therefore, the results prove that our algorithm is able to estimate the path of the route even in intermediate positions of the nodes and deal with the correction of false association of nodes in previous parts of the route.

8. Conclusions

In this paper we have studied the problem of the only-visual topological mapping and route navigation using global appearance image descriptors. First, we have included a comparison of three global appearance techniques and different image sizes. Next, we present the multiscale analysis, which permits estimating the relative position of two images using digital zooming. Then, we include an algorithm to build a topological map from a set of nodes and routes of images. Finally, we have developed a localization algorithm that estimates the position of the mobile in the graph using the visual information as input.

In the comparison of the global appearance descriptors, all the techniques show a reasonable high accuracy in image retrieving tasks. Fourier analysis presents a stable retrieving precision with regard to the images size and a reduced time requirement, but the memory requirement is clearly higher than the other techniques, especially with the bigger images. HOG descriptor shows good computational cost and memory requirements. However, when we reduce the image size, the accuracy in localization decreases more than in the case of the other techniques. Hence, since we pretend to use a reduced image resolution in the map building

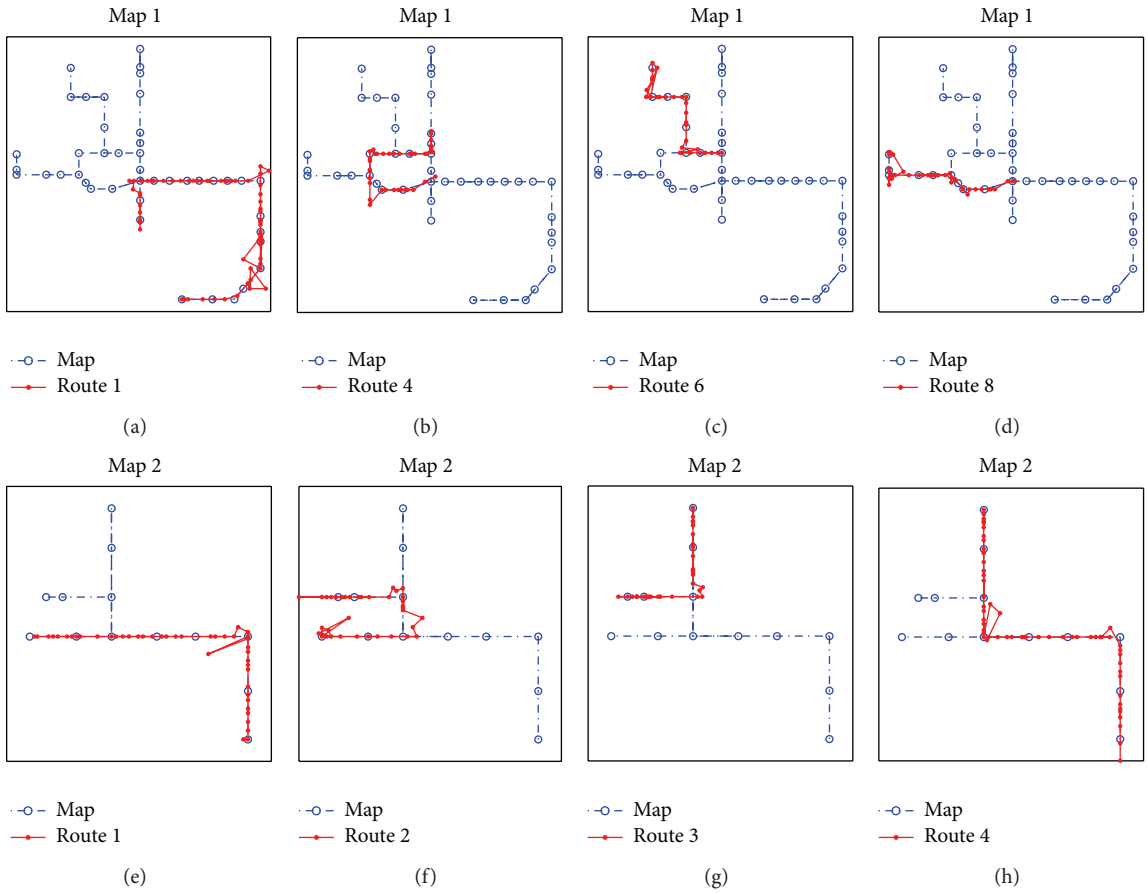


FIGURE 22: Path estimation of the (a) Route 1, (b) Route 4, (c) Route 6, and (d) Route 8 of the Area 1 and path estimation of the (e) Route 1, (f) Route 2, (g) Route 3, and (h) Route 4 of Area 2.

and localization algorithms, HOG is inadvisable in these applications. Gist-Gabor is the most compact representation in almost all the experiments. We select this descriptor to carry out the experiments due to the fact that it is the most reliable descriptor using the lower image resolutions. It is possible to reduce the scene almost 30 times the original size without an important detriment of precision. In this way, the computational time in the image processing in order to obtain the descriptor is reduced more than 10 times.

Regarding the multiscale analysis, it improves the association between images using the global appearance of the scenes and provides a measurement of the topological distance between images.

The map building algorithm is able to determine the adjacency relationships between the nodes distributed in the navigation area and to create a graph using the information of routes taken along the nodes positions. The results present a high accuracy in the node detection and estimation of adjacency and relative orientation. Moreover, the estimation of the topological distance between the nodes provides a graph representation of the nodes with similar layout to the real distribution.

The algorithm created to estimate the path of routes along the area takes advantage of the multiscale analysis to improve the topological localization of the robot in the map. After doing the matching of the route image with the map database, the difference of scales between the node and the route image provides the relative position of both images. Although we use a weighting function in order to penalize important changes in position and orientation between consecutive route images, the algorithm is able to find again the correct location although a previous image of the route would introduce a false pose.

The results obtained both in the map building and the path representations of routes encourage us to continue the possibilities of the application of global appearance image descriptors to these tasks. It would be interesting to extend this study to find the minimum information that the map has to include in order to allow a correct navigation of the robot, the application of new global appearance descriptors, the use of omnidirectional visual information, or the improvement in the estimation of the orientation in order to correct small errors during the navigation.

Conflict of Interests

The authors declare that there is no conflict of interests regarding the publication of this paper.

Acknowledgment

This work has been supported by the Spanish government through the Project DPI2010-15308 "Exploración integrada de entornos mediante robots cooperativos para la creación de mapas 3D visuales y topológicos que puedan ser usados en navegación con 6 grados de libertad."

References

[1] R. Sim and G. Dudek, "Effective exploration strategies for the construction of visual maps," in *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS 2003)*, vol. 4, pp. 3224–3231, October 2003.

[2] T. Camus, D. Coombs, M. Herman, and T.-H. Hong, "Real-time single-workstation obstacle avoidance using only wide-field flow divergence," in *Proceedings of the 13th International Conference on Pattern Recognition*, vol. 3, pp. 323–330, Vienna, Austria, August 1996.

[3] S. Badal, S. Ravela, B. Draper, and A. Hanson, "A practical obstacle detection and avoidance system," in *Proceedings of the 2nd IEEE Workshop on Applications of Computer Vision*, pp. 97–104, Sarasota, Fla, USA, December 1994.

[4] N. Winters, J. Gaspar, G. Lacey, and J. Santos-Victor, "Omni-directional vision for robot navigation," in *Proceedings of the IEEE Workshop On Omnidirectional Vision*, pp. 21–28, 2000.

[5] H. Morita, M. Hild, J. Miura, and Y. Shirai, "Panoramic view-based navigation in outdoor environments based on support vector learning," in *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS '06)*, pp. 2302–2307, Beijing, China, October 2006.

[6] D. G. Lowe, "Distinctive image features from scale-invariant keypoints," *International Journal of Computer Vision*, vol. 60, no. 2, pp. 91–110, 2004.

[7] A. Lingua, D. Marenchino, and F. Nex, "Performance analysis of the sift operator for automatic feature extraction and matching in photogrammetric applications," *Sensors*, vol. 9, no. 5, pp. 3745–3766, 2009.

[8] A. C. Murillo, J. J. Guerrero, and C. Sagüés, "SURF features for efficient robot localization with omnidirectional images," in *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA '07)*, pp. 3901–3907, April 2007.

[9] H. Bay, A. Ess, T. Tuytelaars, and L. van Gool, "Speeded-Up Robust Features (SURF)," *Computer Vision and Image Understanding*, vol. 110, no. 3, pp. 346–359, 2008.

[10] R. Gartshore, A. Aguado, and C. Galambos, "Incremental map building using an occupancy grid for an autonomous monocular robot," in *Proceedings of the 7th International Conference on Control, Automation, Robotics and Vision (ICARC '02)*, vol. 2, pp. 613–618, December 2002.

[11] B. Kröse, R. Bunschoten, S. ten Hagen, B. Terwijn, and N. Vlassis, "Household robots look and learn: environment modeling and localization from an omnidirectional vision system," *IEEE Robotics and Automation Magazine*, vol. 11, no. 4, pp. 45–52, 2004.

[12] E. Menegatti, T. Maeda, and H. Ishiguro, "Image-based memory for robot navigation using properties of omnidirectional images," *Robotics and Autonomous Systems*, vol. 47, no. 4, pp. 251–267, 2004.

[13] I. Kunttu, L. Lepistö, J. Rauhamaa, and A. Visa, "Multiscale fourier descriptor for shape-based image retrieval," in *Proceedings of the 17th International Conference on Pattern Recognition (ICPR '04)*, vol. 2, pp. 765–768, August 2004.

[14] H. Moravec and A. Elfes, "High resolution maps from wide angle sonar," in *Proceedings of the IEEE International Conference on Robotics and Automation*, vol. 2, pp. 116–121, March 1985.

[15] A. Gil, Ó. Reinoso, M. Ballesta, M. Juliá, and L. Payá, "Estimation of visual maps with a robot network equipped with vision sensors," *Sensors*, vol. 10, no. 5, pp. 5209–5232, 2010.

[16] J. Gaspar, N. Winters, and J. Santos-Victor, "Vision-based navigation and environmental representations with an omnidirectional camera," *IEEE Transactions on Robotics and Automation*, vol. 16, no. 6, pp. 890–898, 2000.

[17] N. Winters and J. Santos-Victor, "Omni-directional visual navigation," in *Proceedings of the 7th International Symposium on Intelligent Robotics Systems*, pp. 109–118, 1999.

[18] R. F. Vassallo, H. J. Schneebeli, and J. Santos-Victor, "Visual servoing and appearance for navigation," *Robotics and Autonomous Systems*, vol. 31, no. 1, pp. 87–97, 2000.

[19] J. Košecká, L. Zhou, P. Barber, and Z. Duric, "Qualitative image based localization in indoors environments," in *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, vol. 2, pp. II-3–II-8, June 2003.

[20] A. Štimec, M. Jogan, and A. Leonardis, "Unsupervised learning of a hierarchy of topological maps using omnidirectional images," *International Journal of Pattern Recognition and Artificial Intelligence*, vol. 22, no. 4, pp. 639–665, 2008.

[21] J. Choi, M. Choi, and W. K. Chung, "Topological localization with kidnap recovery using sonar grid map matching in a home environment," *Robotics and Computer-Integrated Manufacturing*, vol. 28, no. 3, pp. 366–374, 2012.

[22] M. Liu, C. Pradalier, F. Pomerleau, and R. Siegwart, "The role of homing in visual topological navigation," in *Proceedings of the 25th IEEE/RSJ International Conference on Robotics and Intelligent Systems (IROS '12)*, pp. 567–572, Vilamoura, Portugal, October 2012.

[23] M. Cummins and P. Newman, "FAB-MAP: probabilistic localization and mapping in the space of appearance," *International Journal of Robotics Research*, vol. 27, no. 6, pp. 647–665, 2008.

[24] M. Cummins and P. Newman, "Appearance-only SLAM at large scale with FAB-MAP 2.0," *International Journal of Robotics Research*, vol. 30, no. 9, pp. 1100–1123, 2011.

[25] N. Bellotto, K. Burn, E. Fletcher, and S. Wermter, "Appearance-based localization for mobile robots using digital zoom and visual compass," *Robotics and Autonomous Systems*, vol. 56, no. 2, pp. 143–156, 2008.

[26] M. Milford, G. Wyeth, and D. Prasser, "Simultaneous localization and mapping from natural landmarks using ratslam," in *Proceedings of the Australasian Conference on Robotics and Automation*, N. Barnes and D. Austin, Eds., Australian Robotics and Automation Association, Canberra, Australia, 2004.

[27] A. J. Glover, W. P. Maddern, M. J. Milford, and G. F. Wyeth, "Fab-map + ratslam: appearance-based slam for multiple times of day," in *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA '10)*, pp. 3507–3512, May 2010.

[28] M. Milford, "Visual route recognition with a handful of bits," in *Proceedings of the Robotics Science and Systems Conference*, N. Roy, Ed., University of Sydney, Sydney, Australia, 2012.

[29] Woodman Labs Inc, 2013, http://gopro.com/hd-hero2-cameras/.

[30] L. Payá, L. Fernández, Ó. Reinoso, A. Gil, and D. Úbeda, "Appearance-based dense maps creation: comparison of compression techniques with panoramic images," in *Proceedings of the 6th International Conference on Informatics in Control, Automation and Robotics*, pp. 250–255, Milan, Italy, July 2009.

[31] F. Amorós, L. Payá, O. Reinoso, and L. M. Jiménez, "Comparison of global appearance techniques applied to visual map building and localization," in *International Conference on Computer Vision Theory and Applications (VISAPP '12)*, vol. 2, pp. 395–398, SciTePress, Science and Technology Publications, Rome, Italy, February 2012.

[32] L. Payá, F. Amorós, L. Fernández, and O. Reinoso, "Performance of global-appearance descriptors in map building and localization using omnidirectional vision," *Sensors*, vol. 14, no. 2, pp. 3033–3064, 2014.

[33] D. Scaramuzza, A. Martinelli, and R. Siegwart, "A flexible technique for accurate omnidirectional camera calibration and structure from motion," in *Proceedings of the 4th IEEE International Conference on Computer Vision Systems (ICVS '06)*, p. 45, January 2006.

[34] N. Dalal and B. Triggs, "Histograms of oriented gradients for human detection," in *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR '05)*, vol. 1, pp. 886–893, June 2005.

[35] A. Friedman, "Framing pictures: the role of knowledge in automatized encoding and memory for gist," *Journal of Experimental Psychology: General*, vol. 108, no. 3, pp. 316–355, 1979.

[36] A. Oliva and A. Torralba, "Modeling the shape of the scene: a holistic representation of the spatial envelope," *International Journal of Computer Vision*, vol. 42, no. 3, pp. 145–175, 2001.

[37] A. Torralba, "Contextual priming for object detection," *International Journal of Computer Vision*, vol. 53, no. 2, pp. 169–191, 2003.

[38] D. Gabor, "Theory of communication. Part III: radio and communication engineering," *Journal of the Institution of Electrical Engineers*, vol. 93, no. 26, pp. 429–457, 1946.

[39] B. S. Manjunath and W. Y. Ma, "Texture features for browsing and retrieval of image data," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 18, no. 8, pp. 837–842, 1996.

[40] A. Gil, O. M. Mozos, M. Ballesta, and O. Reinoso, "A comparative evaluation of interest point detectors and local descriptors for visual SLAM," *Machine Vision and Applications*, vol. 21, no. 6, pp. 905–920, 2010.

[41] D. G. Kendall, "A survey of the statistical theory of shape," *Statistical Science*, vol. 4, no. 2, pp. 87–99, 1989.

[42] I. L. Dryden and K. V. Mardia, *Statistical Shape Analysis*, Wiley Series in Probability and Statistics: Probability and Statistics, John Wiley & Sons, Chichester, UK, 1998.

Hindawi Publishing Corporation
Mathematical Problems in Engineering
Volume 2014, Article ID 678210, 20 pages
http://dx.doi.org/10.1155/2014/678210

Research Article

Cooperative Behaviours with Swarm Intelligence in Multirobot Systems for Safety Inspections in Underground Terrains

Chika Yinka-Banjo,[1] Isaac O. Osunmakinde,[2] and Antoine Bagula[3]

[1] *Department of Computer Science, Faculty of Science, University of Cape Town, Private Bag X3, Rondebosch, Cape Town 7701, South Africa*
[2] *School of Computing, College of Science, Engineering and Technology, University of South Africa (UNISA), P.O. Box 392, Pretoria 0003, South Africa*
[3] *Department of Computer Science, Faculty of Science, University of Western Cape, Private Bag X17, Bellville 7535, South Africa*

Correspondence should be addressed to Chika Yinka-Banjo; chikagog@gmail.com

Received 6 February 2014; Revised 16 May 2014; Accepted 26 May 2014; Published 20 July 2014

Academic Editor: Leo Chen

Underground mining operations are carried out in hazardous environments. To prevent disasters from occurring, as often as they do in underground mines, and to prevent safety routine checkers from disasters during safety inspection checks, multirobots are suggested to do the job of safety inspection rather than human beings and single robots. Multirobots are preferred because the inspection task will be done in the minimum amount of time. This paper proposes a cooperative behaviour for a multirobot system (MRS) to achieve a preentry safety inspection in underground terrains. A hybrid QLACS swarm intelligent model based on Q-Learning (QL) and the Ant Colony System (ACS) was proposed to achieve this cooperative behaviour in MRS. The intelligent model was developed by harnessing the strengths of both QL and ACS algorithms. The ACS optimizes the routes used for each robot while the QL algorithm enhances the cooperation between the autonomous robots. A description of a communicating variation within the QLACS model for cooperative behavioural purposes is presented. The performance of the algorithms in terms of without communication, with communication, computation time, path costs, and the number of robots used was evaluated by using a simulation approach. Simulation results show achieved cooperative behaviour between robots.

1. Introduction

Multirobot systems share the need to cooperate, creating the problem of modelling behaviour. When dealing with multiple robots, with randomness involved, the dynamic and unpredicted nature of the environment has to be considered. Hence, the behavioural modelling system has to cope with the random (dynamic and unpredictable) nature of the system. Researchers, on the other hand, have been captivated by this cooperative and coordinated problem of multirobot systems (MRS) in recent times. A list of literature on multiple robots' cooperation implemented in space was reviewed in [1]. Using multiple robots to achieve tasks has been more effective than using a single robot. See for instance [2, 3] (and all references therein) for some specific robotic tasks.

For a full discussion of underground mines, benefits, and disaster rates, see [4]. Safety is a major element in the underground mine; despite a significant reduction through safety research measures, the number of disasters in underground mines in South Africa and the world at large remains high [5]. To contribute to underground mine safety, that is, to prevent disasters from occurring, interacting autonomous multirobots are sent before mine operations resume to inspect how safe the underground mine is. This is achieved by assessing the status of the rocks, roofs, and level of toxic gases.

A multirobot planning algorithm for tunnel and corridor environments is implemented in [6]. The overall problem formulation is implemented using a topological graph and spanning representation. Activities, such as a single robot

drive and differential robots drive that can rotate in place, are carried out at different positions of the graph. Different methods have been used to tackle coordination of MRS in different domains [7]. Complete task coordination by multirobots was handled [3, 8]. An extension of a market-based approach was used. This was achieved by generalizing task descriptions into tasks trees, thereby allowing tasks to be traded in a market setting at a variable level of abstraction. Figure 1 shows an overview of the inspection environment.

We consider a scenario in which an MRS cooperates to achieve a common goal of safety inspection in a partially observable environment such as the underground terrain. Each robot requires to be guided using the proposed cooperative behavioural model. Finding a plan to achieve this cooperative model often involves solving an NP-hard problem. This is so because it involves multiple interacting robots. The interaction which comes as a result of minimizing time involved in achieving underground inspection requires reasoning among the robots. In this case, we use the QL technique to cleverly achieve the reasoning aspect of this work and combine it with optimal route finding using ACS. The two major questions answered in this problem are the following: (1) which state/room should I inspect now without repetition and a collision with another robot? (2) How do I get there using the closest link? However, all robots have partial knowledge of the environment and they are all equipped with the necessary sensors required for the inspection. The major contributions of this paper are as follows:

(i) development of a hybrid QLACS swarm intelligent model based on Q-Learning and the ant colony system for addressing cooperative behaviours in MRS achieving underground terrain safety inspections;

(ii) detailed experimental evaluations of the proposed QLACS model conducted on a simulated publicly available underground tunnel. Also, the proposed model is benchmarked with related methods;

(iii) systematic description of worked scenarios on an optimal route finder and MRS cooperative inspection using QLACS for ease of implementation by system engineers, robotics researchers, and practitioners.

We do not know of any study that analytically describes how a swarm intelligent model can easily be implemented and applied to a multirobot system in a heterogeneous environment. In the next section, we discuss related work in the area of multirobot systems for underground terrains' safety, reinforcement learning (RL), and ant colony optimization (ACO). We then detail our proposed cooperative behaviour framework and approach. The experimental results and evaluations of the safety inspections follow. Finally, we conclude by summarizing and discussing other future extensions.

2. Theoretical Background

In this section, we review some of the related work to MRS for underground terrains safety, which is the foundational domain for our research. A look at RL algorithms and its paradigms follows in Section 2.1. ACO and its paradigm applied in MRS conclude Section 2.

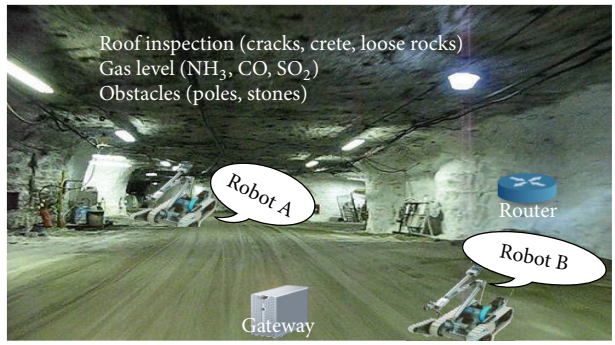


Figure 1: Overview of MRS inspection in an underground mine.

2.1. Related Multirobot Systems for Underground Terrains Safety. Effective, exhaustive, quick, and safe navigation of an MRS is subject to its ability to perceive, interpret, and interact (cooperate) within their environment. MRS can be used to achieve different tasks autonomously in structured and unstructured environments. In these environments, the automation of the operations is effected in different areas of MRS research: biologically inspired robot teams, communication, architectures and task planning, localization and mapping, object transportation and manipulation, learning, and so forth [9]. However, an environment such as the underground terrain has been a grey application domain in MRS research.

As indicated in Section 1, Peasgood et al. [6] implemented a multiphase algorithm for multirobot planning in tunnel environments. The algorithm as presented assumed a centralized planning architecture. They further compared the multirobot planning with a sequential planner. Their future work was to consider a decentralized planner architecture and might explore hybridizing the two planning algorithms.

Thorp and Durrant-Whyte discussed a starter on field robots [10]. From their discussion, field robotics involves the automation of platforms such as air, sea, and land in harsh unstructured environments such as underwater exploration, mining, agriculture, and highways. Field robots are made up of three parts: navigation and sensing, planning and control, and safety. Their work also discussed the challenges and progress of field robots. One of the major challenges of field robots in harsh environments such as an underground mine is the problem of position determination (localization). This is so because the global positioning system (GPS) can only help in an environment where the sky views are guaranteed. However, progress has been made in automating some of the environments that are cooperatively constrained.

The proposed model presented in [4] promised to handle the safety part of field robots presented by Thorp and Durrant-Whyte [10] in underground mines. Their model architecture had three layers; the first layer handles the cooperative behaviour of the model, the second layer deals with the scalability degree of the model, and the last layer

Steps:
initialize $Q(s, a)$ *arbitrarily*
repeat (for each episode):
initialize s
Repeat (for each step of episode):
Choose a *from* s *using policy derived from* Q
Take action a, *observe* s, s'

$$Q(s,a) \leftarrow Q(s,a) + \alpha\left[r + \gamma \max_{a'} Q(s', a') - Q(s,a)\right]$$

$s \leftarrow s'$
until s *is terminal*

Algorithm 1: Q-learning algorithm.

handles the applicability of the model. This paper is building on what has already been proposed in [4].

An investigation into automating the underground mine environment after blasting, called "making safe," was carried out in [11] to ensure the safety of the environment after blasting. Blasting in an underground mine is the controlled use of explosives to excavate, break down, or remove rock. The need to investigate the stability of the environment after blasting before any mining operation takes place is of the highest priority, hence, the reason for automation. The automation was centred on a persistent area of concern in South African underground mine operation called hanging walls which is caused as a result of rock burst and fall of ground. There are also other persistence areas such as the levels of toxic gases which pose great disaster threats to the lives of miners, for instance, heat sicknesses, explosions, pneumoconiosis (occupational safety and health-fall of ground management in South Africa, SAMRASS-code book for mines), and so forth. Some of these disasters might result in fatalities and/or disabilities. Again, when an accident happens during mining operations, rescuers find it difficult to respond immediately to accidents. Looking at the aforementioned concerns, there will be a need to create models for safety inspection of underground mine operations. For instance, monitoring the underground mine environment for detecting hazardous gases and/or smoke should be one of the important safety measures. Continuous monitoring of workers and equipment is another crucial safety measure [5]. Picking up the sound from roof cracking to monitor when a roof is about to fall is also a safety item.

2.2. Reinforcement Learning. For a robot to operate in a harsh unstructured environment, considering every possible event in defining its behaviour is intricate [12]. It is, however, essential to develop robots that can conform to changes in their environment. RL is one of the artificial intelligence (AI) algorithms that can achieve learning by experience. This enhances robots' interaction with the environment. We investigate the use of RL to assist the behaviours of an MRS in safety inspection of underground mines. RL is a sequence of actions and state transitions with some associated rewards. What is being learned in RL is an optimal policy (what is the right thing to do in any of these states (s)) [13].

At any time step each robot is in a specific state in relation to the environment and can take one of the following actions: inspect, ignore, or shutdown. Each robot receives feedback after performing an action, which explains the impact of the action in relation to achieving the goal. The effect of the action can be either a good or bad reward. This reward is measured in terms of values. Therefore, the value of taking an action a in any state s of the underground terrain is measured using the *Action-Value function* called Q-value $Q^{\pi}(s, a)$. When a robot is starting from state s, taking action a, and using a policy pi(π), an expected return which is defined as the sum of the discounted rewards is achieved.

In this research, Q-learning, a method of RL, is explored. The purpose of RL methods is to study $Q^{\pi}(s, a)$ values so as to achieve optimal actions in the states. QL is an online RL method that requires no model for its application and stores the reinforcement values outcome in a look-up table. The QL architecture used in this work consists of learning threads, which amount to the number of robots involved in the inspection behavioural task. Each robot in the learning thread carries out Q-learning in the environment. Algorithm 1 explains the QL algorithm used in the behavioural model.

α is the learning rate set between 0 and 1. At 0, Q-values are never updated, hence nothing is learned; learning can occur quickly at 1. γ is the discount rate set between 0 and 1 as well. This models the fact that the future rewards are worth less than the immediate rewards. $\max_{a'}$ is the maximum reward that is attainable in the state following the current state. That means the reward for taking the optimal action thereafter.

QL is a competitive and search-based algorithm inspired by computational theory. It is not necessarily a multiagent algorithm but can be adapted to a multiagent or multigoal scenario. The success of this algorithm relies on the value and policy iterations, which can be adjusted by some unfairness (heuristics) to fit the current problem scenario. The most competitive action is selected by its value and action leads to another state or condition. Both value and policy are updated after each decision. Harnessing QL for an MRS scenario increases the cost exponentially and the overall performance drops in the same direction. As the robots increase cost, such as completion time, memory usage, and awareness factor

Table 1: Variant of ACO.

S/N	Year	Algorithm
1	1991	Ant system (AS)
2	1992	Elitist A.S
3	1995	Ant-Q
4	1996	Ant colony system
5	1996	Max-Min A.S (MMAS)
6	1997	Ranked based A.S
7	1999	ANTS
8	2000	BWAS
9	2001	Hypercube A.S

(other robots in the environment), search time increases. However, following our heuristic model of QL which was mainly determined by the policy we set to achieve our goal, our QL performs well above traditional QL.

2.3. Ant Colony Optimization. ACO is a type of swarm intelligence (SI). Bonabeau et al. [14] defined SI as any attempt to design algorithms or distributed problem-solving devices inspired by collective behaviour of social insect colonies and other animal societies. This implies that anytime something is inspired by swarms, it is called swarm intelligence. Researchers have been thrilled by swarms because of some of their fascinating features. For instance, the coordinated manner in which insect colonies work, notwithstanding having no single member of the swarm in control, the coordinated ways in which termites build giant structures and how the flocks move as one body, and the harmonized ways in which ants quickly and efficiently search for food can only be attributed to an emergent phenomenon [15, 16].

Ants, an example of a social insect colony, achieve their self-organizing, robust, and flexible nature not by central control but by stigmergy. Stigmergy, also known as a pheromone trail, describes the indirect communication that exists within the ant's colony. The indirect communication which is triggered by pheromone trails helps in recruitment of more ants to the system. Ants also use pheromones to find the shortest paths to food sources. Pheromone evaporation prevents stagnation, which also helps to avoid premature convergence on a less than optimal path [17].

ACO is necessarily a multiagent based algorithm. The first implementation of this optimization algorithm is in a classical search based (combinatory) problem, the travelling salesman problem, giving the shortest path to a specified destination. After this feat, many researchers have used it or its variants to model different problems. In this work, a variant of ACO is used to find the optimal path for MRS. Table 1 describes variants of ACO and their meaning [18]; see also [19] and all references therein.

In AS, the pheromones are updated by all the ants that complete a tour. ACS is the modified version of AS, which introduces the pseudorandom proportional rule. In elitist AS, ants that belong to the edges of the global best tour get an additional amount of pheromone during the pheromone update. MMAS introduces an upper and lower bound to the values of the pheromones trails. All the solutions are ranked to conform to the ants' length in ranked-based AS [20].

Our interest is in implementing ACS to find the best possible round trip inspection path in our model environment. This best possible inspection path will be supplied as input or made available for the robots using QL for inspection decisions. In most of the route finding scenarios, the nodes are linearly joined and are not largely distributed. This is a scenario where ants can forage along at least two paths (multigoal path environment, though in our case there is a path exposed to four different goals; see Section 3).

3. Proposed Cooperative MRS Behaviour Framework

The following are some of the factors we are considering in the course of building the model. Each robot has to learn to adjust to the unknown underground mine environment. In this case, each robot requires intelligence and a suitable machine learning algorithm is adopted. No robot has global information of the environment because of its limited sensing capabilities. Each robot has limited communication capabilities; therefore, each robot has to keep track of the information of others to remain in a team. Figure 2 describes the proposed framework as an approach for building the distributed, coordinated inspecting model for MRS.

The framework indicates three layers of the model: the bottom layer, the middle layer, and the topmost layer. The learning capability of the MRS is achieved in the bottom layer, with the reinforcement learning algorithm and swarm intelligence technique. This intelligence enables robot A to take action knowing the action of robot B and vice versa. At the middle layer, scalability in terms of the number of robots the system can accommodate is achieved. This is expedient because a team of robots tends to achieve tasks more quickly and effectively than single robots. This scalability is handled with some memory management techniques, as indicated in Figure 2. The real life implementation is achieved by using the information acquired from the topmost layer. Figure 3 is the breakdown of the framework in Figure 2.

There is a base-station or server that serves as a backup for the information captured and analysed from individual robots. The model proposed in this research deals with the way in which robots need to find their way within communication range and uses a broadcast approach for effective communication of navigation status. A team of robots cooperatively inspecting an area in the underground mine will need to know where they are and where to go next, so it is obviously a continuing problem and our contribution in this case is that before robot R1 takes an action, it broadcasts its location and inspection status to other robots, R2, R3, and so forth, and vice versa. An unreachable robot receives packets of information based on the destination address through rerouting from the nearer robots. The reliability of this broadcast method is the ability to determine the extent of the task executed already by looking at the memory of any leading robot in the team.

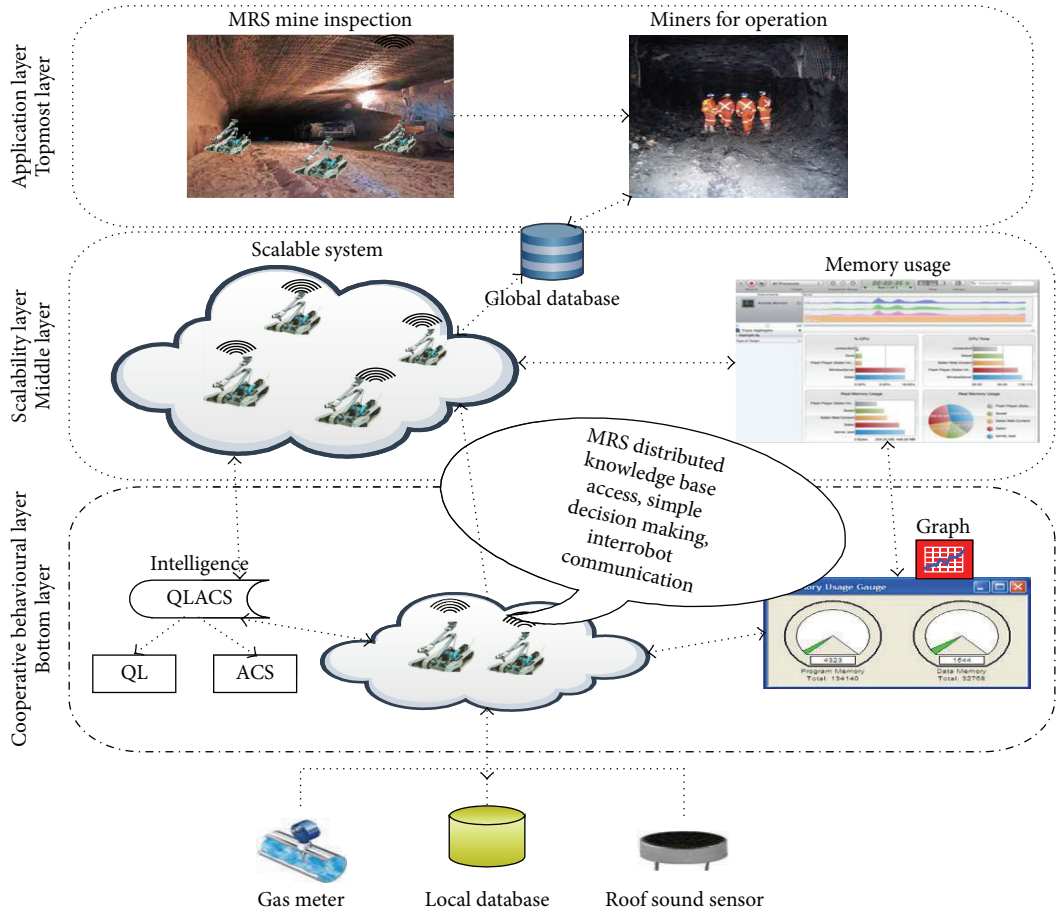


Figure 2: Framework of the proposed QLACS model for cooperative behaviours.

Table 2: State and possible actions of the environment.

	State	Possible Actions
1	*A* (lower left part (LLP))	Inspect, ignore
2	*B* (lower middle part (LMP))	Inspect, ignore
3	*C* (lower right part (LRP))	Inspect, ignore
4	*D* (middle left part (MLP))	Inspect, ignore
5	*E* (central part (MCP))	Inspect, ignore
6	*F* (upper left part (ULP))	Inspect, ignore
7	*G* (upper right part (URP))	Inspect, ignore
8	*H* (outside mine part (OMP))	Shutdown

3.1. Problem Formulations. Suppose we have seven rooms/states connected by doors/links representing underground mine regions as shown in Figure 4 and labeled as shown in Table 2. We label each room *A* through *F*. The outside of the mine can be thought of as one big room (*H*). Notice that doors *F* and *C* lead outside the mine *H*. We put two robots in rooms *F* and *C* as their starting states, respectively. The robots inspect one state at a time, considering the obstacles encountered in the process. The robots can change direction freely, having links/doors to other rooms/states. In each of the states in Figure 4 two actions are possible: inspection of roof cracks (RC) and level of toxic gases (TG), or ignoring the state, as it has been inspected earlier. According to our proposed model, a robot can inspect at least half of the given underground mine region to attain its maximum performance, which in turn attracts a good reward. When the current state of a robot is the end point of the inspection task, the robots exit the mine using the exit points *C* and *F*, respectively. The possible transition states of the robots are displayed using the state diagram in Figure 5. The state diagram and the transition matrix are formed using the QL algorithm.

The global map is assumed to be known by the robots but there is no prior knowledge of the local map. Robots only know what they have sensed themselves and what their teammates communicate to them. Not only does our improved QL depend on the map of the environment but also each robot learns through experience about local changes. They explore and inspect from state to state until they get to their goal states. Our proposed model QLACS achieves an offline mode for the route finding algorithm (ACS). This means that the global view of the map would have been provided before the learning robots start.

The model is explained using simulations; the results are presented in Section 4. Some of the results also have graphical interpretations. Analysis by a state transition diagram is

Table 3: Initial reward matrix.

		Robot's action							
		A	*B*	*C*	*D*	*E*	*F*	*G*	*H*
Robot's state	*A*	—	50, 100	—	50, 100	—	—	—	—
	B	50, 100	—	50, 100	—	50, 100	—	—	—
	C	—	50, 100	—	—	—	—	50, 100	150
	D	50, 100	—	—	—	50, 100	50, 100	—	—
	E	—	50, 100	—	50, 100	—	50, 100	50, 100	—
	F	—	—	—	50, 100	50, 100	—	50, 100	150
	G	—	—	50, 100	—	50, 100	50, 100	—	—
	H	—	—	50, 100	—	—	50, 100	—	—

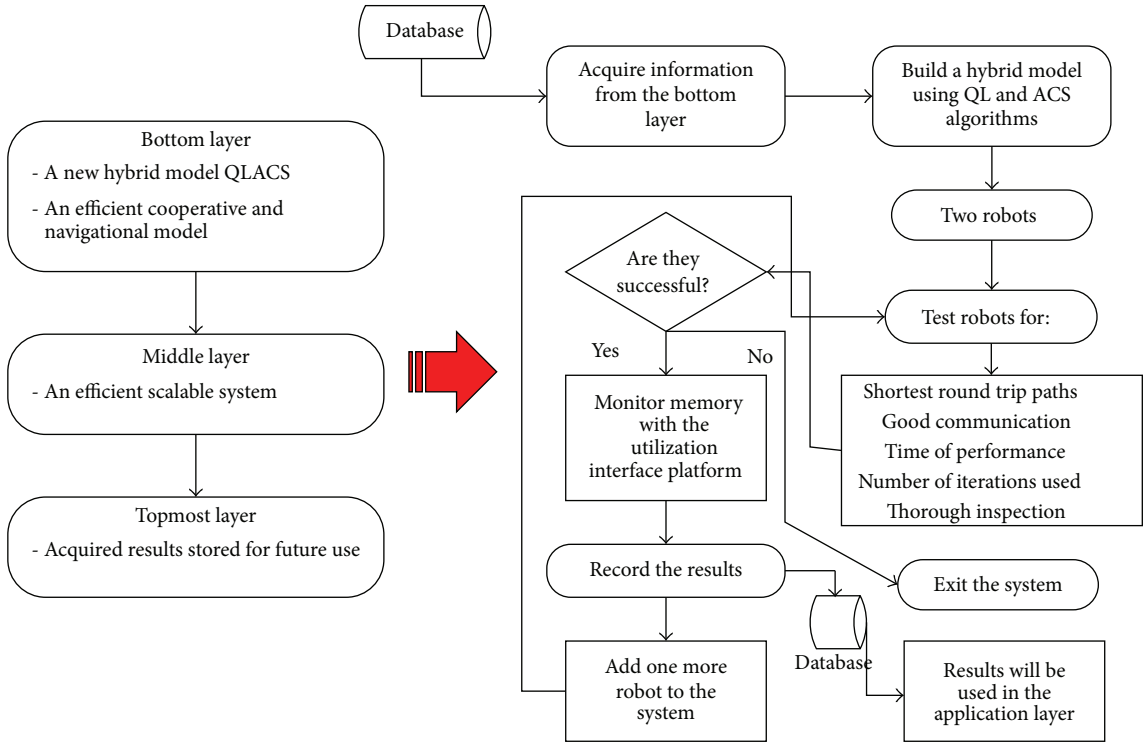


Figure 3: Breakdown of the framework. (a) Contributions of the framework, Layer by layer, and (b) processes of the multirobot behavioural system.

presented in Figure 5. The possible state actions of the robots are presented in Table 3. The states of the robots are reduced as follows: searching or inspecting for roof cracks and toxic gas levels in each state or room and recording the outcome of the inspection in the robots' memories. The processes involved in achieving good communication while the robots inspect the states are broadcasting inspected states to the other robots and ignoring the inspected/broadcasted state, avoiding collision with obstacles and other robots and finally moving to the next available state that has not been inspected.

The Q-learning algorithm is used to determine what action is to be selected in any particular state. Let the action (a) = inspect, ignore, shutdown, state space (s) = dimensions in the topological map, sensor readings (s_r), hazardous conditions (H_c) = roof crack (RC), toxic gas level (TG). Each robot is configured with an aerial scanner and chemical sensitive sensor that will provide readings (s_r), to determine if there is a hazardous condition, H_c, in the environment. The selection of an action by a robot through the Q-learning algorithm is based on the information from the broadcast. The whole framework uses a decentralized Q-learning scheme such that each robot has its own thread with its Q-learning and table. All Q-values on the tables of the robots in the team are initialized to zero; that is, states corresponding to such positions have not been inspected. When a robot enters a new state it broadcast its information for the current state to all other robots in the environment. The broadcast indicates whether the current state has been

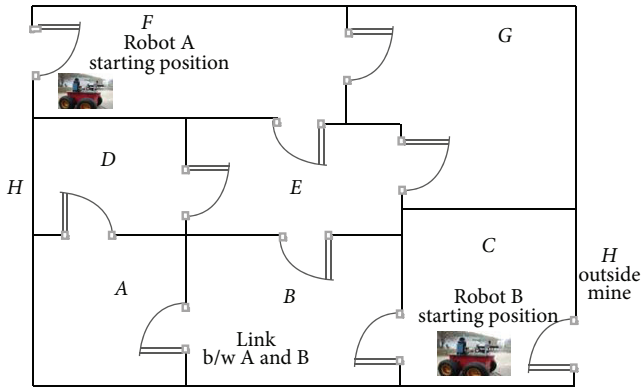


Figure 4: Model of the environment with two entrances and exits (2EE).

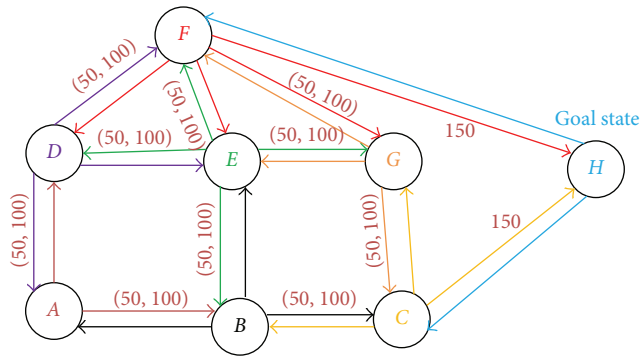


Figure 5: Events and transition diagram of the modeled environment with H as goal state.

inspected or not. The broadcast is a search of corresponding positions in the memories of all visible robots. If the resultant search returns a zero; the broadcast indicates that the state has not been inspected and if it returns a value greater than zero it indicates that the state is not available for inspection. All robots must receive a broadcast before they act in any particular state. When a robot gets to a state, it receives a broadcast and passes its value to the QL algorithm to make a decision. The robot carries out the decision of the QL algorithm. The policy of the Q-learning algorithm makes a robot carry out an inspection if the resultant search of the broadcast returns zero and ignores it if the resultant search is greater than zero. A robot only shuts down if all states have been visited and inspected. As the broadcast information is passed to the Q-learning algorithm, the policy is iterated towards an optimal value and condition.

The broadcast avoids the cost of multiple inspections and the QL ensures that robots take appropriate actions. The only state in which inspection is carried out would have sensor readings (s_r), indicating H_c. For taking an action (a) in state (s), the robot gets a reward (R), which is used in (10) to compute the Q-value for the current state and can send a broadcast to other robots. Figure 5 and Table 3 show the restrictions and possible transitions among node points, indicating the possible rewards for any particular action (a) in any state (s). Every other transition besides the goal state could result in a reward of 50 or 100 and at completion the reward is the maximum, 150. We consider it wasteful to make the robots scan first before sharing intelligence because such action would slow them down and make them expend more energy. Robots are to provide the inspection results, showing actions taken in different states and the nature of conditions detected in any particular state. The introduction of the broadcast approach to determine the team's exploration reduces the execution time and energy cost of the whole team and makes the collaboration effective. So in a multirobot scenario the task can be effectively executed if the robots are made to share intelligence as they progress. Robots do not have to waste time and energy in doing the same task already carried out by other robots in the team.

GIVEN. On a mathematical justification of the above, suppose a safety preinspection of toxic gases or rock fall or some combination of the two is being carried out on a piece of complex underground terrain in Figure 1, say $L\,\mathrm{Km}^2$; there is a limited number of MRS with different capacities, R, and precious inspection time, T minutes. Every region/state x_1 in the terrain requires a capacity of robot R_1 of MRS and limited time T_1 while every state x_n requires robot R_n of MRS and inspection time, T_n. Let P_1 be the positive reward of QL for correct behaviour on state x_1 and let P_n be the reward on state x_n. This research aims to maximise positive rewards by choosing optimal values for states $x_1, \ldots, x_n$ as follows:

Maximise:

$$P_1 \cdot x_1 + P_2 \cdot x_2 + \cdots + P_n \cdot x_n$$

(objective function)

Subject to constraints:

$$x_1 + x_2 + \cdots + x_n \le L$$

(limited total states)

$$R_1 \cdot x_1 + R_2 \cdot x_2 + \cdots + R_n \cdot x_n \le R \tag{1}$$

(limited MRS capacity)

$$T_1 \cdot x_1 + T_2 \cdot x_2 + \cdots + T_n \cdot x_n \le T$$

(limited inspection time)

Non-negativity constraints:

$$x_1 \ge 0, \quad x_2 \ge 0, \ldots, x_n \ge 0$$

(MRS cannot inspect negative states).

In terms of solving this optimization problem, we use the proposed QLACS model to compare the time and number of states inspected. The number of robots used is also compared. The graphs results in Section 4 also give more insight into the solution of the problem.

3.2. Basic Navigation and Cooperative Behaviours Using QLACS. QLACS has two components. The first component is formed by an improved ACS and the second component

is formed by an improved QL. The improvement occurs because some heuristics were added to the ordinary QL and ACS to achieve the hybrid QLACS. However, the second component of QLACS, which is an improved QL, was initially used to solve the proposed problem. After much analysis, we realized that the system needs to be optimized for effective cooperation and communication.

Using the second component of QLACS to solve the basic navigation and cooperative behaviour, the possible actions were set for each robot as inspect, ignore, and shutdown (after reaching the goal state H). Also, a reward system that will reflect the possible actions of the robots was chosen. In other words, a robot gets 150 points only when the goal is achieved (shutdown), 100 points for ignoring an already inspected area (ignore), and 50 points for inspecting an uninspected area (inspect). Figure 5 shows the transition events of the model and Table 3 displays the possible state action for each robot. The way the QLACS second component works here is based on both navigation and communication behaviours.

Achieving navigational behaviour with the second component of QLACS has some cost associated to it. In our scenario, because we want the robots to share intelligence by broadcasting results (robots search through other robots' memory), our problem is not solely navigational but also cooperative. Our behavioural actions are inspect, ignore, and shutdown. We noted that these actions of interest are not navigation oriented; there is no way we could use them in making decisions on transition. The functions for decision can be integrated to assist the other. Therefore, our behavioural model is an integration of two behaviours: (1) navigational behaviour and (2) cooperating behaviour through decision making. The integration works with a route finding method called *RandomStateSelector,* which we introduced in the second component of QLACS to help determine where the robot goes from the starting point to the exit point. Two parts of the *RandomStateSelector* method are introduced in this work. The first one is the *RandomStateSelector_C_H,* which is used to transit from state C to H and the second one, *RandomStateSelector_F_H,* transits from state F to H. This method works but not effectively because some of the states are repeated several times because of the random selection method. However, the decision part of this second component of QLACS, which is handled by a method called *CheckInspection,* worked efficiently. *CheckInspection* is responsible for sharing the broadcast among the agents. The broadcast looks at all the stored Q-values on all the robots and returns a signal for the action that needs to be taken. Therefore, we concluded that the heuristically accelerated component of QLACS has proven to be effective by showing evidence of effective communication in making inspection decisions using our model. It did not guarantee shortest possible time for inspection because of repeated states decisions. In this light, we only harnessed the communication strength of the second component of QLACS for communication and cooperation. Figure 5 and Table 3 form the basis for the QLACS second component.

To take care of the random state selector problem encountered in implementing the algorithm used for the second part of QLACS, we introduced an optimized route finder algorithm. This route finder algorithm, which forms the first component of QLACS, is a swarm intelligence technique. Figure 6 and Table 4 form the basis of the exploration space of the agents (ants) which achieved the optimum route finding. The weighted graph in Figure 6 is based on the number of obstacles the ants will encounter from the points of entry F and C. The combined table in Table 4 contains the weights of the obstacles and evidence of an edge between any two vertices (states). It shows that there is a connection between any two vertices in a graph. Generally, a "1" or "2" depicts the existence of an edge while a "0" represents the absence of an edge, that is, no transition between such vertices. The constructed graph is an undirected multigraph, providing evidence of some agents coming from F or C of the mine (logical space). It is unidirectional because agents can move in any particular direction (multigraph). This means that the same weights on the edges apply to both directions. The graph does not show H; we assume that once the agents reach F or C, they exit if all inspections have been done.

Table 4: Combined adjacency and weight matrix.

	F	G	E	D	A	B	C	H
F	0	1	1	1	0	0	0	1
G	1	0	2	0	0	0	1	0
E	1	2	0	2	0	2	0	0
D	1	0	2	0	2	0	0	0
A	0	0	0	2	0	2	0	0
B	0	0	2	0	2	0	1	0
C	0	1	0	0	0	1	0	1
H	1	0	0	0	0	0	1	0

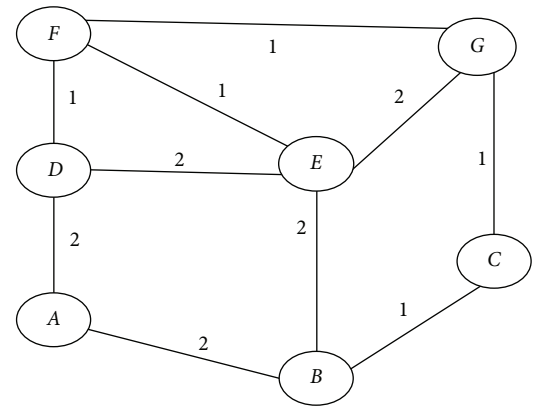


Figure 6: Weighted map/graph of the model environment.

3.3. Hybrid Model Development . The parameters for building the analytical hybrid QLACS (equations (2) through (11)) model are presented in Tables 5 and 6.

3.3.1. Algorithmic and Mathematical Analysis

ACS Starts. Computation of edge attractiveness

$$\eta_{i,j} = \frac{1}{D_{i,j}}. \tag{2}$$

Table 5: A list of parameters for the ACS model.

ACO parameters	Meaning
α	Pheromone influence factor
β	Influence of adjacent node distance
ρ	Pheromone evaporation coefficient
Q	Attractiveness constant
e	Visited edge
e'	Edge not visited
L_k	Length tour of ant k
τ	Pheromone concentration (amount)
η	Specific visibility function (attractiveness)
$\Delta\tau^k$	Pheromone concentration by Kth ant
$P_r(i, j)$	Probability of moving from i to j
$D_{i,j}$	Visibility or distance between i and j
f_i	Fitness of individual in a population
P_i	Probability of being selected among f_i
N	Number of individuals in the population
i, j	Denotes any two adjacent nodes in the graph
M_k	Set of unvisited nodes

Table 6: A list of parameters for the QL model.

QL Parameters	Meaning
Q	Q-value update
s	State
a	Action
R	Reward
γ	Learning rate

Computation of instantaneous pheromone by ant k

$$\Delta\tau^k = \frac{Q}{L_k}. \tag{3}$$

Update of pheromone

$$\tau_{i,j} = (1-\rho) * \tau_{i,j} + \Delta\tau^k_{i,j}. \tag{4}$$

Computation of edge probability

$$P_r(i, j) = \frac{[\tau_{i,j}]^\alpha [\eta_{i,j}]^\beta}{\Sigma_{e'=(i,j)} [\tau_{i,j}]^\alpha [\eta_{i,j}]^\beta}. \tag{5}$$

Adoption of roulette wheel

$$\text{Cumulative}\left(P_r(i, j)\right) = \sum_{i=1}^{N+1} P_r(i, j), \tag{6}$$

$$f_i = \frac{\sum_{j=1}^{N} f_j}{N}, \tag{7}$$

$$P_i = \frac{f_i}{\sum_{j=1}^{N} f_j}. \tag{8}$$

Equations (2) through (8) build the complete route finding model. Equations (2) through (4) are prerequisite to (5). Equation (5) is prerequisite to roulette wheel selection. At the end of (8), new states are selected and the trail is updated. The best path from both directions is selected and used as input in Q-learning. Note that $D_{i,j}$ = weight on edges, $P_r(i, j)$ = chance of moving to a node $E(I, j)$, L_k = sum of visited nodes by ant k.

QL Starts. Each robot in its QL thread computes its learning rate:

$$\gamma = \frac{0.5}{\left[1 + \text{Frequency}(s, a)\right]}. \tag{9}$$

Q-values are updated:

$$Q(s, a) = R(s, a) + \gamma \tag{10}$$

making a broadcast (Decision = Inspect/Ignore/Shutdown)

$$Q(s, a) = \begin{cases} Q = 0 & \text{Inspect if } s_j \neq \text{goalstate} \\ Q > 0 & \text{Ignore if } s_j \neq \text{goalstate} \\ Q \geq 0 & \text{Shutdown if } s_j = \text{goalstate}. \end{cases} \tag{11}$$

Equation (9) is Gamma, the learning rate, which is always between zero and 1. This equation is calculated based on the frequency of action of each robot in inspecting states. Equations (9) to (11) are state-dependent. The states are kept in a buffer and then accessed at run time. ACS and QL do not work simultaneously. ACS works to completion and QL takes the final output as its input. ACS is not repeatedly called while QL is working.

Our behavioural model is an integration of two algorithms: (1) route finding algorithm (2) communication and cooperative algorithm. The integration works the following way. The optimal route finder (ACS) determines where the robot goes from the starting point to the destination, while QL determines what action it takes when it gets to any of the states. This collaboration works effectively because the optimal route finder has been proven to give the best possible transitions and the heuristically accelerated QL has proven to be effective by showing evidence of effective communication in making inspection decisions. Consequently, it guarantees the shortest possible time for inspection in the absence of wasteful inspection decisions. This framework forms the basis for our cooperative behaviour model for MRS (QLACS). The pseudocode for the implementation of QLACS is outlined in Algorithm 2.

3.3.2. QLACS Hybrid Approach Evolution. Figure 7 is the architecture of hybrid QLACS explaining the handshake between the two components.

(i) Graph Construct Module. This is the interconnection of states in our model environment (see Figure 4). It encompasses all possible transitions from F to C and vice versa. Thus from the model we construct an adjacency/weight graph matrix that can be traversed by any graph-oriented algorithm.

INPUT: Edge distance(obstacles), pheromones, ants' trail, associated probabilities, starting and terminating indexes that is, from *F* or *C*
OUTPUT: Effective cooperation, inspection and navigation

(1) ***Boolean*** CompletedFlag = ***False*** //Boolean variable indicates completion for all the threads
(2) ***Declare*** CompletedThreadBuffer //Data structure stores Info about completed thread
(3) Initialize all Qvalues to ***Zero*** //All Qvalues positions are initialized to zero
(4) Initialize Best Paths From ACO algorithm //starting from *F* and *C*
(5) ***While (CompletedFlag*** <> ***True)*** //Checks for when all robots have finished and flag is true
Begin
Create *N* number of Threads in Parallel
Threads get Current States in Parallel from ACO algorithm
Threads get Next States and Indexes in Parallel from ACO algorithm
Threads compare Qvalues of all corresponding positions of Current States (Each Robot Broadcast Qvalue info)
IF ((Q == 0) & (ThreadNextState <> ***GoalState))*** //Checks if a particulate state is available
Begin
State is Available, Robot with the CurrentThreadID Inspects
Compute and Update Qvalue
End
IF (Q > 0) //checks if a state is not available, because an already inspected state has $Q > 0$
Begin
State is already inspected, Robot with the CurrentThreadID Ignore
Compute and Update Qvalue
End
IF ((Q == 0) & (ThreadNextState == GoalState)) //Checks for goal state and shuts down.
Begin
Compute and Update Qvalue
Goal state is reached and Inspection Completed
Thread with the CurrentThreadID Shuts down
Store CurrentThreadID ***in CompletedThreadBuffer***
End
IF (Count [CompletedThreadBuffer] == NumberOfRobot) //Learning stops when this happens
Begin
CompletedFlag = True
End
End of While Loop.

Algorithm 2: Pseudocode for QLACS.

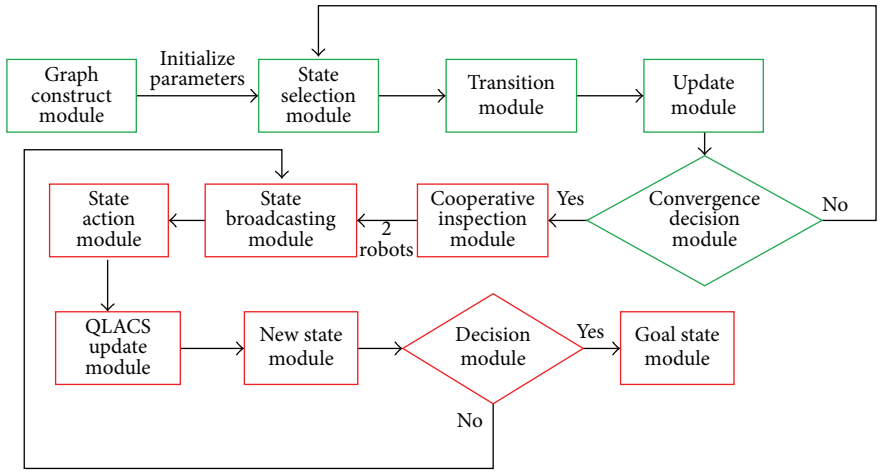


Figure 7: Hybrid architecture.

Example I of QLACS (Using Figure 5 for optimized route finding)
Parameters used in the first component of QLACS
$Q = 2.0, \alpha = 3, \beta = 2, \rho = 0.01$
Starting State: F
Terminating State: C/F
Terminating condition: When all states have been visited at least once and it is at terminating state
State Space = $\{F, G, E, D, A, B, C\}$
Initialize pheromones positions to 0.01
Rand = (0, 1) returns a random value between 0 and 1
Equations
(i) Computation of attractiveness $\eta_{i,j}$ using (2)
(ii) Computation of instantaneous pheromones by ant k for all potential states using (3)
(iii) Pheromone update using (4)
(iv) Computation of probabilities for the potential states using (5)
(v) Adaptation of roulette Wheel using (6)
Equation (5) can be reduced to Let $w(i, j) = [\tau_{i,j}]^{\alpha}[\eta_{i,j}]^{\beta}$

$$\text{Sum} = \sum_{i=1}^{N+!} w(i, j)$$

$$\text{So } P_r(i, j) = \frac{w}{\text{sum}}$$

Algorithm 3: Parameters for QLACS Example I.

In our case, there are eight states: primarily seven internal states and a common terminating state. Since the states are not just linear, the environment allows for multiple options; that is, an agent/ant can choose from any present state. This type of scenario is also called a multigoal scenario.

(ii) State Selection Module. Here, the agents select the start and end states, which according to our model in Figure 4 are F and C. These states are selected based on the cumulative probability of two adjacent nodes in (6).

(iii) Transition Module. This module takes care of the transition rules of the agents by calculating the pheromone concentrations, distances, and probabilities using (2) through (4).

(iv) Update Module. After transition from one state to another, an update of the pheromone is computed, after which multipath planning with shortest path is achieved.

(v) Convergence Decision Module. This is where the best trail/path decision is taken. This is the end product of the first component of QLACS, which is then moved to the second component of QLACS for cooperative behaviour.

(vi) Cooperative Inspection Module. This is where the robots are deployed to start inspection. The robots are to use the acquired best paths starting from F and C, respectively, as input for the second component of QLACS. The two robots are to use the learning rate from (9) to learn the environment and use (10) for cooperation.

(vii) State Broadcasting Module. This module handles the broadcasting behaviours of the two robots, which are achieved by using (10). Each robot checks its memory represented by Q-values before taking any decision.

(viii) State Action Module. State broadcasting by each robot is immediately followed by action selection. In other words, the state to inspect or ignore is achieved here using (11).

(ix) QLACS Update Module. After each action selection, the Q-values of each robot are updated using (10).

(x) New State Module. This module takes care of the robot's new state after updating the Q-values. Each robot runs in its own threads, managing its memory, yet sharing information.

(xi) Final Decision Module. This decision module determines if the robot should exit the environment or still do more inspections. It is also controlled by (11).

(xii) Goal State Module. The completion of the second component of QLACS is getting to the goal state after successful inspection of states. This goal state according to our model in Figure 4 is called H.

3.3.3. Analytical Scenario. For the benefit of robotic and system engineers, practitioners, and researchers, this paper uniquely presents scenarios on the ease of implementation of the proposed QLACS as described in Algorithms 3 and 4 and Tables 7 and 10. The first component of QLACS is explained in Example I, shown in Algorithm 3 and Table 7. The first component that achieved optimal route paths for the robots has the parameters listed in Algorithm 3. The calculation for different states transition for the first component of QLACS is analysed in Table 7.

Repeat steps 1–6 for subsequent current states until the termination condition and state are reached. At the end of seven updates we have Tables 8 and 9. The total length shown in Table 8 represents the total number of obstacles encountered by each agent while trailing to achieve the optimal route for the robots. The number of obstacle shown

Table 7: QLACS Example I navigational analytical solution.

Starting state: F
Potential states: D, E, G
(1) Use (2), $\eta_{F,D} = 1$, $\eta_{F,E} = 1$, $\eta_{F,G} = 1$
(2) Use (3), $L_k = 1$, $\Delta\tau^k(F,D) = 2/1 = 2$, $\Delta\tau^k(F,E) = 2$, $\Delta\tau^k(F,G) = 2$
(3) Use (4), $\tau_{F,D} = (1 - 0.01) * 0.01 + 2 = 2.0099 = 2.01$, $\tau_{F,E} = 2.01$, $\tau_{F,G} = 2.01$
First pheromone update

F	G	E	D	A	B	C
0.01	2.01	2.01	2.01	0.01	0.01	0.01

(4) Use (5) $w(F,D) = [\tau_{F,D}]^{\alpha}[\eta_{F,D}]^{\beta} = (2.01)^3 = 8.12$, $w(F,E) = 8.12$, $w(F,G) = 8.12$
Sum $= w(F,D) + w(F,E) + w(F,G) = 24.36$
$P_r(F,D) = \dfrac{w}{\text{sum}} = \dfrac{8.12}{24.36} = 0.33$, $P_r(F,E) = 0.33$, $P_r(F,G) = 0.33$
Probabilities

F	G	E	D	A	B	C
0	0.33	0.33	0.33	0	0	0

(5) Use (6)

H	F	G	E	D	A	B	C
0	0	0.33	0.33	0.33	0	0	0

Call Rand
Rand = 0.07, Rand falls in state E. Roulette wheel selects E as the next state, end of roulette wheel.
(6) Update trail: F, E

Current state: E
Potential states: B, D, G
(1) Use (2), $\eta_{E,B} = 1/2$, $\eta_{E,D} = 1/2$, $\eta_{E,G} = 1$
(2) Use (3), $L_k = (F - E - D) = 1 + 2 = 3$, $L_k = (F - E - B) = 1 + 2 = 3$, $L_k = (F - E - G) = 1 + 2 = 3$
$\Delta\tau^k = \dfrac{2}{3} = 0.67$
(3) Use (4), $\tau_{E,B} = (1 - 0.01) * 2.01 + 0.67 = 2.66$, $\tau_{E,D} = 2.66$, $\tau_{E,G} = 2.66$
Second pheromone update

F	G	E	D	A	B	C
0.01	2.66	2.01	2.66	0.01	2.66	0.01

(4) Use (5) $w(E,B) = (2.66)^3 * (1/2)^2 = 4.71$, $w(E,D) = 4.71$, $w(E,G) = (2.66)^3 * (1)^2 = 18.82$
Sum $= 4.71 + 4.71 + 18.82 = 28.24$
$P_r(E,B) = \dfrac{w}{\text{sum}} = \dfrac{4.71}{28.24} = 0.17$, $P_r(E,D) = \dfrac{4.71}{28.24} = 0.17$, $P_r(E,G) = \dfrac{18.82}{28.24} = 0.67$
Probabilities

F	G	E	D	A	B	C
0	0.67	0	0.17	0	0.17	0

(5) Use (6)

H	F	G	E	D	A	B	C
0	0	0.67	0	0.17	0	0.17	0

Call Rand
Rand = 0.9, Rand falls in state D. Roulette wheel selects D as the next state, end of roulette wheel.
(6) Update trail: F, E, D

Example II of QLACS (For good cooperation and communication between robots)
Parameters used in the second component of QLACS (Using output from the first component of QLACS)
Reward Scheme: Inspect = 50, Ignore = 100, Shutdown = 150
State space: Optimized path from QLACS_R1 = {*F*, *E*, *D*, *A*, *B*, *C*, *G*, *C*} and QLACS_R2 = {*C*, *B*, *A*, *D*, *F*, *E*, *G*, *C*}
Starting State: *F*/*C*
S_j = Terminating State: *C* then *H*
Terminating condition: When all states have been visited.
Initialize *Q*value positions to zeros
Equations
(i) Compute learning rate using (9)
(ii) Compute update on $Q(s, a)$ using (10)

Algorithm 4: Parameters for QLACS Example II.

Table 8: Pheromone update of a full cycle.

	Pheromone update							
Current states	*F*	*G*	*E*	*D*	*A*	*B*	*C*	
F	0.01	2.01	2.01	2.01	0.01	0.01	0.01	1st update
E	0.01	2.66	2.01	2.66	0.01	2.66	0.01	2nd update
D	3.13	0.01	3.03	0.01	3.03	0.01	0.01	3rd update
A	0.01	0.01	0.01	0.45	0.01	0.45	0.01	4th update
B	0.01	0.01	0.67	0.01	0.67	0.01	0.7	5th update
C	0.01	0.91	0.01	0.01	0.01	0.89	0.01	6th update
G	0.71	0.01	0.69	0.01	0.01	0.01	0.71	7th update

C = terminating state and ant *k* has moved through all states at least once.
Trail: *FEDABCGC*.
Number of obstacles = 1 + 2 + 2 + 2 + 1 + 1 + 1 + 1 = 10.

Table 9: Probability update of a full cycle.

	Probability table						
Current states	*F*	*G*	*E*	*D*	*A*	*B*	*C*
F	0	0.33	0.33	0.33	0	0	0
E	0	0.67	0	0.17	0	0.17	0
D	0.69	0	0.16	0	0.16	0	0
A	0	0	0	0.5	0	0.5	0
B	0	0	0.16	0	0.16	0	0.68
C	0	0.52	0	0	0	0.49	0
G	0.45	0	0	0	0	0	0.45

C = terminating state.

as 10 is calculated using the trail result in Table 8. Table 9 displays the probability update table for a full cycle of route finding. In the case of this first example, the first robot will terminate its inspection through *C* and then *H*.

Algorithm 4 displays the parameters of the second example of QLACS. The second component of QLACS handles this aspect of the model, which is the communication and cooperative part. Once the first component hands the output to the second component, it becomes the second component input and it runs with it. From Algorithm 4, QLACS_R1 represents the input for robot 1 and QLACS_R2 represents the input for robot 2 received from the first component of QLACS. The terminating condition is when all states have been visited.

The rest of Example II displayed in Table 10 explains the cooperative behavioural scenario from one state to another for two robots. The tables in the last row of Table 10 show the communication and cooperative output achieved using the second component of QLACS.

4. Experimental Evaluations: Safety Inspections for MRS Behaviours

The experimental results of implementing QLACS in an environment that consists of obstacles and links are tabulated in this section. The experimental setup is explained in Section 4.1. Different performance categories are shown in this section: without communication category and with communication category. In the without communication category, as displayed in Section 4.2, we found that robots can inspect all states individually without knowing that another robot exists. Robots can also inspect some of the states, thereby leaving some states not inspected. The communication category is explained in Section 4.3 while the performance of the QLACS measured with other existing methods is tabulated in Section 4.4.

4.1. Experimental Setup. Figure 8 displays different sets of experiments conducted in the environment using two robots. Figure 8(a) shows how two robots resume inspection from two different entrances. In each set of experiments, the robots take the readings of the roof cracks and level of toxic gases using their sensors. The same behaviours happen in Figures 8(b) and 8(c), respectively, at different inspection locations. The inspection locations vary in the four experimental setups shown in Figure 8.

4.2. Experiment 1: Performance of QLACS without Cooperation. The result of implementing the QLACS without communication in the proposed environment (Figures 1 and 4) is shown in Tables 11 and 12. In the case of Table 11, robot 1 (R1), enters the mine through state *F* while robot 2 (R2) enters the mine through state *C*. However, each robot learns by inspecting some of the states and ignoring some of the

Table 10: QLACS Example II cooperative behaviour.

Starting simulation Robot 1 **Starting state: *F*** (1) Use (9) $\gamma(F) = \frac{0.5}{[1+1]} = \frac{0.5}{2} = 0.25$ (2) Check the Q-value for state F (use (11)) $Q(F, a) = 0$ Selected action, a = inspect (3) use (10) $Q(F, a) = 50 + 0.25(0) = 50$ End of value iteration	**Current state: *E*, Robot 1** (1) Use (9) $\gamma(E) = \frac{0.5}{[1+1]} = \frac{0.5}{2} = 0.25$ (2) Check the Q-value for state E (Use (11)) $Q(E, a) = 0$ Selected action, a = inspect (3) Use (10) $Q(E, a) = 50 + 0.25\,(\max(0, 0, 0)) = 50$ End of value iteration	**Current state: *C*, Robot 2** (1) Use (9) $\gamma(C) = \frac{0.5}{[1+1]} = \frac{0.5}{2} = 0.25$ (2) Check the Q-value for state C (use (11)) $Q(C, a) = 0$ Selected action, a = inspect (3) Use (10) $Q(C, a) = 50 + 0.25\,(\max(0, 0, 0)) = 50$ End of value iteration
Current state: *B*, Robot 2 (1) Use (9) $\gamma(B) = \frac{0.5}{[1+1]} = \frac{0.5}{2} = 0.25$ (2) Check the Q-value for state B (use (11)) $Q(B, a) = 0$ Selected action, a = inspect (3) Use (10) $Q(B, a) = 50 + 0.25\,(\max(0, 0, 0)) = 50$ End of value iteration	**Current state: *D*, Robot 1** (1) Use (9) $\gamma(D) = \frac{0.5}{[1+1]} = \frac{0.5}{2} = 0.25$ (2) Broadcast (use (11)) $Q(D, a) = 0$ Selected action, a = inspect (3) Use (10) $Q(D, a) = 50 + 0.25\,(\max(0, 0, 0)) = 50$ End of value iteration	**Current State: *A*, Robot 2** (1) Use (9) $\gamma(A) = \frac{0.5}{[1+1]} = \frac{0.5}{2} = 0.25$ (2) Broadcast (use (11)) $Q(A, a) = 0$ Selected action, a = inspect (3) Use (10) $Q(A, a) = 50 + 0.25\,(\max(0, 0, 0)) = 50$ End of value iteration
Current state: *A*, Robot 1 (1) Use (9) $\gamma(A) = \frac{0.5}{[1+0.25]} = \frac{0.5}{1.25} = 0.4$ (2) Broadcast (use (11)) $Q(A, a) = 50$, that is, $Q > 0$ Selected action, a = Ignore (3) Use (10) $Q(A, a) = 100 + 0.4\,(\max(0, 0, 0)) = 100$ End of value iteration	**Current state: *D*, Robot 2** (1) Use (9) $\gamma(D) = \frac{0.5}{[1+0.25]} = \frac{0.5}{1.25} = 0.4$ (2) Broadcast (use (11)) $Q(D, a) = 50$, that is, $Q > 0$ Selected action, a = ignore (3) Use (10) $Q(D, a) = 100 + 0.4\,(\max(0, 0, 0)) = 100$ End of value iteration	**Current state: *B*, Robot 1** (1) Use (9) $\gamma(B) = \frac{0.5}{[1+0.25]} = \frac{0.5}{1.25} = 0.4$ (2) Broadcast (use (11)) $Q(B, a) = 50$, that is, $Q > 0$ Selected action, a = Ignore (3) Use (10) $Q(B, a) = 100 + 0.4\,(\max(0, 0, 0)) = 100$ End of value iteration
Current state: *F*, Robot 2 (1) Use (9) $\gamma(D) = \frac{0.5}{[1+0.25]} = \frac{0.5}{1.25} = 0.4$ (2) Broadcast (use (11)) $Q(F, a) = 50$, that is, $Q > 0$ Selected action, a = ignore (3) Use (10) $Q(F, a) = 100 + 0.4\,(\max(0, 0, 0)) = 100$ End of value iteration	**Current state: *C*, Robot 1** (1) Use (9) $\gamma(B) = \frac{0.5}{[1+0.25]} = \frac{0.5}{1.25} = 0.4$ (2) Broadcast (use (11)) $Q(B, a) = 50$, that is, $Q > 0$ Selected action, a = Ignore (3) Use (10) $Q(B, a) = 100 + 0.4\,(\max(0, 0, 0)) = 100$ End of value iteration	**Current state: *E*, Robot 2** (1) Use (9) $\gamma(E) = \frac{0.5}{[1+0.25]} = \frac{0.5}{1.25} = 0.4$ (2) Broadcast (use (11)) $Q(E, a) = 50$, that is, $Q > 0$ Selected action, a = ignore (3) Use (10) $Q(E, a) = 100 + 0.4\,(\max(0, 0, 0)) = 100$ End of value iteration
Current state: *G*, Robot 1 (1) Use (9) $\gamma(G) = \frac{0.5}{[1+1]} = \frac{0.5}{2} = 0.25$ (2) Broadcast (Use (11)) $Q(G, a) = 0$ Selected action, a = Inspect (3) Use (10) $Q(G, a) = 50 + 0.25\,(\max(0, 0, 0)) = 50$ End of value iteration	**Current state: *G*, Robot 2** (1) Use (9) $\gamma(G) = \frac{0.5}{[1+0.25]} = \frac{0.5}{1.25} = 0.4$ (2) Broadcast (use (11)) $Q(G, a) = 50$, that is, $Q > 0$ Selected action, a = ignore (3) Use (10) $Q(G, a) = 100 + 0.4\,(\max(50, 50, 50))$ $= 100 + 20 = 120$ End of value iteration	**Current state: *C*, Robot 1** (1) Use (9) $\gamma(C) = \frac{0.5}{[1+0.4]} = \frac{0.5}{1.4} = 0.36$ (2) Broadcast (Use (11)) $Q(C, a) = 50$ Selected action, a = Ignore (3) Use (10) $Q(C, a) = 100 + 0.36\,(\max(50, 50))$ $= 100 + 18 = 118$ End of value iteration
Current state: *C*, Robot 2 (1) Use (9) $\gamma(C) = \frac{0.5}{[1+0.36]} = \frac{0.5}{1.36} = 0.37$ (2) Broadcast (Use (11)) $Q(C, a) = 50$, that is, $Q > 0$ Selected action, a = ignore (3) Use (10) $Q(C, a) = 100 + 0.37\,(\max(50, 50))$ $= 100 + 0.37 * 50 = 118.5$ End of value iteration	**All QLACS 1 states exhausted** **Goal state: *H*, Robot 1** (1) Use (9) $\gamma(H) = \frac{0.5}{[1+1]} = \frac{0.5}{2} = 0.25$ (2) Broadcast (use (11)) $Q(H, a) = 0$ Selected action, a = Shutdown (3) Use (10) $Q(H, a) = 150 + 0.25\,(\max(0, 0)) = 150$ End of value iteration	**All QLACS 2 states exhausted** **Goal state: *H*, Robot 2** (4) Use (9) $\gamma(H) = \frac{0.5}{[1+1]} = \frac{0.5}{2} = 0.25$ (5) Broadcast (use (11)) $Q(H, a) = 0$ Selected action, a = Shutdown (6) Use (10) $Q(H, a) = 150 + 0.25\,(\max(0, 0)) = 150$ End of value iteration

Table 10: Continued.

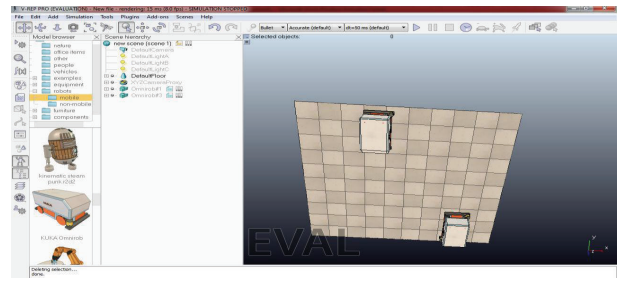
	Robot 1				Robot 2			
		Inspect	Ignore	Shutdown		Inspect	Ignore	Shutdown
End of policy iteration	*F*	Yes	No	No	*C*	Yes	No	No
	E	Yes	No	No	*B*	Yes	No	no
	D	Yes	No	No	*A*	Yes	No	No
	A	No	Yes	No	*D*	No	Yes	No
	B	No	Yes	No	*F*	No	Yes	No
	C	No	Yes	No	*E*	No	Yes	No
	G	Yes	No	No	*G*	No	Yes	no
	C	No	Yes	Yes through *H*	*C*	No	Yes	Yes through *H*

states. Since there is no communication, they exit the mine after learning, consequently not inspecting all the states. The same routine is reflected in Table 12, but in this case, each robot ends up inspecting all the states before exiting the mine. Analysis of Tables 11 and 12 shows that resources are wasted and the inspection job is not effective and efficient. Comparing the two tables, the time and distance cost are high, though higher in Table 12 because of many repetitions. For instance, the time and distance cost in Table 12 column 2 are 48.0028 and ((*F, G, E, D, A, B, C*), (*C, B, A, D, E, G, F*)), respectively. It also shows that the states are repeated in Tables 11 and 12. The memory usage is equally high. This led us to create a more efficient QLACS with communication by introducing some heuristics. The processes involved in Tables 11 and 12 are explained in Table 13.

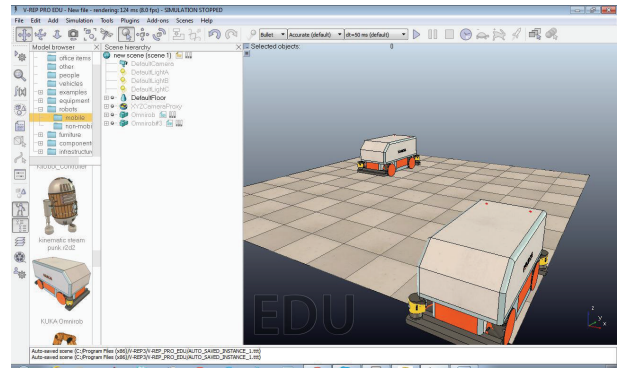
One can see from Tables 11 and 12 that there is no good communication among the robots, hence the evolvement of Table 14.

4.3. Experiment 2: Performance of QLACS with Good Cooperation. The heuristic added to the known QL made this experiment show good communication. This is where our contribution to communication is shown. As a robot inspects and learns the environment, it broadcast its Q-values to the other robot, which is in the form of a lookup table. In this case, each robot checks for Q-values; when a Q-value is equal to zero; the robot randomly selects a state for inspection. If a Q-value is greater than zero, the robot checks if the state is available for inspection or for ignoring. When a robot encounters a state with a Q-value equal to zero and the thread next to the state is equal to the goal state (H), then it shuts down. It must have checked the lookup table to see that all states have been inspected. The result in Table 14 shows good communication between two robots. No states were inspected more than once. The iterations for every run with their times, memory usage, and effective communication are also displayed in Table 14.

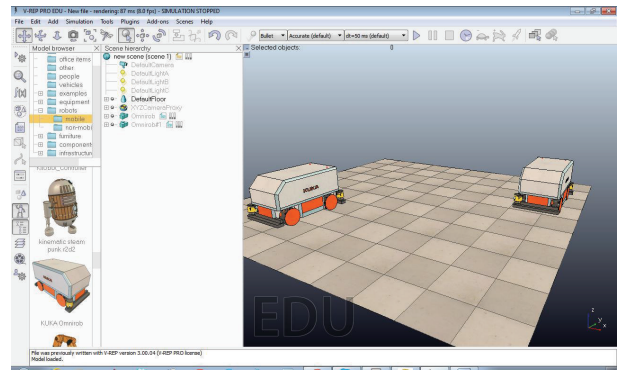
Comparing Table 14 with Tables 11 and 12, one cannot but notice the huge differences in the memory, time, and distance costs. The communication between the robots in Table 14 resulted in achieving good inspection; however, the random method of choosing next inspection states did not give an optimized route, thereby increasing time cost in passing and checking through inspected states.



(a) Two robots starting inspection from two different entrances



(b) Two robots inspecting as they navigate in the environment



(c) Two robots inspecting different locations with different positions

Figure 8: Different experimental behaviours for two robots.

4.4. Experiment 3: Performance of QLACS for the Navigation Behaviour of the Proposed Model. The result of implementing the first component of QLACS for effective navigation of the proposed model is tabulated in Table 16. Table 15 shows

Table 11: QLACS without communication (inspecting some states).

Number of runs	1	2	3	4	5	6	7
Iterations	9	10	10	13	13	9	13
Time (sec)	43.0025	30.0017	34.002	30.0017	31.0017	31.0018	27.0016
Memory usage (bytes)	18872	18160	19204	18208	17896	18164	18308
Inspected states (R1)	*G*	*F*, *G*, *C*	*C*	*F*, *D*, *B*	*E*, *A*	*F*, *E*, *D*, *A*, *C*	*F*, *G*, *E*, *A*
Inspected states (R2)	*B*, *A*, *G*	*C*	*C*, *B*, *A*, *E*, *F*	*B*, *A*	*C*, *B*, *E*	*B*, *A*, *E*	*A*, *G*

Table 12: QLACS without communication (inspecting all states).

Number of runs	1	2	3	4	5	6	7
Iterations	114	10	70	10	9	10	10
Time (sec)	43.0024	48.0028	41.0023	34.0019	34.0019	34.0019	35.002
Memory usage (bytes)	28440	17960	27216	17000	18456	17672	17968
Inspected states (R1)	*F, G, E, D, A, B, C*	*F, G, E, D, A, B, C*	*F, G, E, D, A, B, C*	*F, G, E, D, A, B, C*	*F, G, E, D, A, B, C*	*F, G, E, D, A, B, C*	*F, G, E, D, A, B, C*
Inspected states (R2)	*C, B, A, D, E, G, F*	*C, B, A, D, E, G, F*	*C, B, A, D, E, G, F*	*C, B, A, D, E, G, F*	*C, B, A, D, E, G, F*	*C, B, A, D, E, G, F*	*C, B, A, D, E, G, F*

Table 13: Processes used in achieving Tables 11 and 12.

(a) For Table 11

Inspecting some States
(1) Initialize the starting and goal states
(2) Initialize all Qvalues to zeroes
(3) Select a random action If Qvalues of all possible actions at current state are zeroes
(4) Select the action with the highest Qvalue If it is the Max Qvalue
(5) Compute and update Qvalue of the selected action
(6) Get new state among possible states
(7) If **new state = goal state** then go to **Steps 5 and 9**
(8) Repeat **Steps 3 to 6** until **Step 7**
(9) Shutdown

(b) For Table 12

Inspecting all States
(1) Initialize the starting and goal states
(2) Initialize all Qvalues to zeroes
(3) Select a random action If Qvalues of all possible actions at current state are zeroes
(4) Select the action with the highest Qvalue If it is the Max Qvalue
(5) Compute and update Qvalue of the selected action
(6) Get new state among possible states
(7) If **new state = goal** state then go to **Steps 5 and 10**
(8) Repeat **Steps 3 to 7** until **Step 9**
(9) All states except the goal state has taken **Action = Inspect**
(10) Shutdown

the selected parameters used in achieving the optimal path. The optimal path found after nine test runs is the path with a distance cost of 10 for both entries to the environment, displayed in Table 16. Therefore, columns 4, 5, 7, 8, and 9 can be used as the optimized path input for the first component of QLACS. Then QLACS will use any of the optimized paths to navigate to the specified states and take decisions accordingly. For instance, the test run result for nine ants gives 16 iterations under 60.0035 seconds and giving the shortest paths to both robots coming from entries *F* and *C* of the environment. The path that gives us cost as 10 is obtained from FEDABCGC (1.2.2.2.1.1.1) and CBADFGEGC (1.2.2.1.1.2.2.1), respectively.

Alpha (α), Beta (β), and Rho (ρ) represent the heuristic properties of the ACS algorithm. The Alpha factor is the measure of the influence of pheromone concentration that can influence the selection of the path with the highest pheromone concentration. Beta is the measure of the influence which is related to the distance between any two adjacent nodes. It is also a heuristic factor that can measure the influence distance in selecting the next state. It is not limited to pheromones. Rho has to do with the rate at which the pheromones evaporate. Rho shows how often new paths are explored by the agents/ants rather than reinforcing old paths. Each agent cooperates by having access to other agents' pheromone values. The pheromone value is initialized to 0.01 and it is continually updated until learning stops. It is similar to the first component of QLACS, where we initialize all QLACS positions to zero and update for every new step.

Table 16 gave the optimized route to be used by the robots to achieve inspection tasks. This resulted in combining Tables 14 and 16 to achieve Table 17. Table 17 shows optimized time cost, memory usage, route cost, and good communication.

4.5. Experiment 4: Benchmarking the New Hybrid Model (QLACS) with Popular Methods. Choosing the optimized paths, QLACS performs cooperative inspection behaviour through communication among robots. Looking at the sample result on Table 17, QLACS chooses the shortest and complete possible inspection routes from different runs. In this case, the best paths were obtained from Table 16 by using 9 agents, 10 agents, and 12 agents. All the test runs gave best trail paths from both entrances listing the complete states

Table 14: QLACS with communication.

Number of runs	1	2	3	4	5	6	7
Iterations	9	10	9	13	10	10	9
Time (sec)	33.0019	31.0017	31.0018	31.0018	29.0023	30.0017	31.0018
Memory usage (bytes)	15748	15948	15748	18232	16576	15948	15748
Inspected states (R1)	*F*, *G*, *E*	*F*, *G*, *E*, *D*	*F*, *G*, *E*	*F*, *E*	*F*, *G*, *E*, *D*, *B*	*F*, *G*, *E*, *D*	*F*, *G*, *E*
Inspected states (R2)	*C*, *B*, *A*, *D*	*C*, *B*, *A*	*C*, *B*, *A*, *D*	*F*, *G*, *E*, *D*, *A*	*C*, *A*	*C*, *B*, *A*	*C*, *B*, *A*, *D*

Table 15: Navigation behaviour parameter specification.

ACO properties	Type of ACO	Population	Length of path	Pheromone coefficient β	Heuristic coefficient α	Evaporation rate ρ
Properties	ACS	9 or 12	8	2	3	0.01

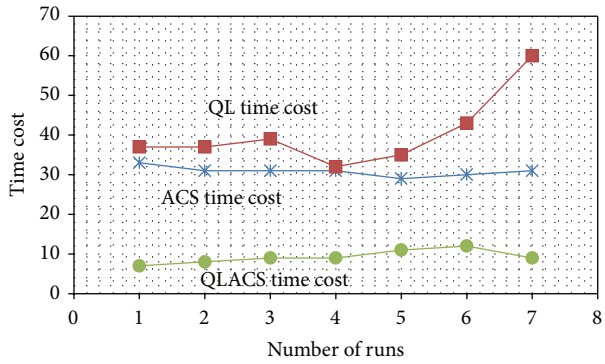


Figure 9: Comparison of time costs for QL, ACS, and QLACS.

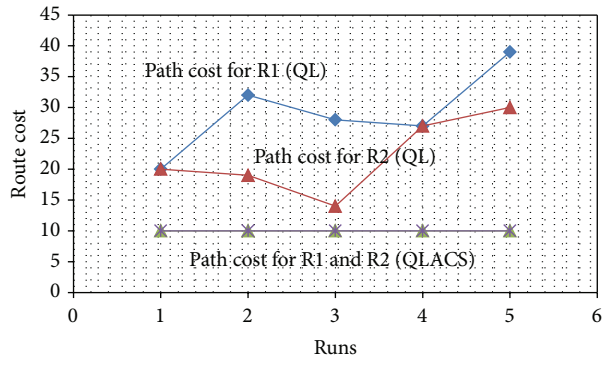


Figure 10: Comparison of route costs for QL and QLACS.

with length 10 as shown in Table 16; that is, they have states from *F* to *C* and from *C* to *F*. The length is the addition of weights, along the line (trail edges). Then the QLACS uses the optimized paths to make decisions on inspections. The result from Table 17 shows that no state is inspected twice or visited more than required. The QLACS model concentrates on making the best inspection decisions for MRS. The first run on Table 17 shows that nine agents/ants were used for the route finding, the optimized route was achieved after nine iterations under 7.0004 sec, and the states where all inspected effectively without redundancies.

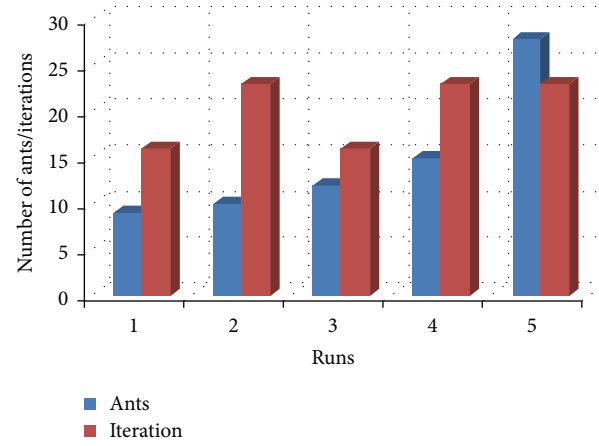


Figure 11: Number of ants/iteration required for QLACS to find optimal path.

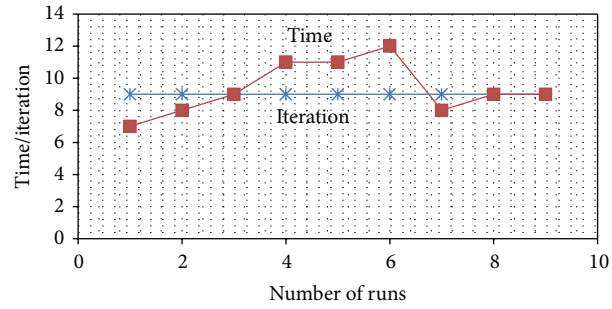


Figure 12: Time and iteration required for QLACS to achieve optimal behavior.

4.5.1. Comparative Study of the Proposed Model (QLACS) with QL Only and ACS Only. Based on the results tabulated in Table 18 and Figure 9, the average time costs of achieving the MRS behaviours for QL, ACS, and QLACS were compared. Two robots will use an average of 30.8590 sec to achieve thorough inspection behaviour using QL, an average of 40.4309 sec to achieve optimal navigation behaviour using ACS, and an average of 9.2868 sec to achieve both navigation and cooperative behaviour using QLACS. The result shows

Table 16: Navigation behaviour computations.

Number of runs	1	2	3	4	5	6	7	8	9
Number of Ants	5	7	8	9	10	11	12	15	28
Iterations	12	26	14	16	23	14	16	23	22
Time (sec)	37.0021	37.0021	39.0022	60.0035	32.0019	33.0019	35.002	43.0025	65.0037
Best train distance (R1)	12	12	10	10	10	10	10	10	10
Best trail distance (R2)	10	12	12	10	10	12	10	10	10

Table 17: New hybrid model QLACS computations.

Number of runs	1	2	3	4	5	6	7	8	9
Iterations	9	9	9	9	9	9	9	9	9
Number of ants	9	9	9	10	10	10	12	12	12
Time (sec)	7.0004	8.0005	9.0005	11.0006	11.0006	12.0007	8.0005	9.0005	9.0006
Memory usage (bytes)	10288	10280	10248	10288	10288	10288	10164	10712	10164
Inspected states (R1)	F, G, E, D	F, G, E, D	F, G, E, D	F, G, E, A, B	F, G, E, A, B	F, G, E, A, B	F, G, D, A	F, G, D, A	F, G, D, A
Inspected states (R2)	C, B, A	C, B, A	C, B, A	C, D	C, D	C, D	C, B, E	C, B, E	C, B, E

Table 18: Time cost comparison for QL, ACS, and QLACS.

Runs	QL time cost (sec)	ACS time cost (sec)	QLACS time cost (sec)
1	33.0019	37.0021	7.0004
2	31.0019	37.0021	8.004
3	31.0018	39.0022	9.0005
4	31.0018	32.0019	9.0005
5	29.0023	35.002	11.0006
6	30.0017	43.0025	12.0007
7	31.0018	60.0035	9.0006
Average	30.8590	40.4309	9.2868

that our proposed integrated algorithm performs better with reduced time cost. On the same note, the route costs for the QL and QLACS were also compared. The results in Table 19 and Figure 10 show that the proposed model QLACS gave a much lower route cost than the QL.

The number of ants and iterations used to achieve this is displayed in Figure 11. The more the ants, the more the iteration. The best routes created in the five test runs shown in Figure 11 are run numbers 1 and 3. They used fewer ants and iterations to achieve the optimal routes. The optimal result for the proposed model is achieved under nine iterations for every run. The iteration remains constant for any number of agents and robots. The blue line with star markers at Figure 12 is the iteration value for 9 runs. The red line shows the different amounts of time required for each run. The time is also almost stable for the QLACS, though it fluctuated a little in the middle. This signifies that the proposed model is more robust and effective in finding the optimal route and coordinating MRS. The reduced route cost and shorter computation time achieved with the QLACS satisfied the criteria for cooperative behavioural purposes.

4.6. Experiment 4: Scalability of QLACS Using Two, Three, and Four Robots. In this section, we present some results obtained by experimenting with the cooperative behavioural action of two, three, and four robots using the QLACS model. The performance of the two, three, and four robots was evaluated by running the simulation three times, using the same number of agents. The performance of the proposed QLACS model shows good communication between the two, three, and four robots under the states inspected, in the last four columns of Table 20. An almost stable time was used in achieving the inspection for all the robots. The detail of the simulation result is laid out in Table 20.

A notable observation emanating from Table 20 is that there is a need for a larger mine area of inspection because robot R1 in rows 4 to 6 could inspect only one state, while robots R1 and R2 in rows 7 to 9 could inspect only one state each. This implies that the size of the field of inspection is proportional to the number of robots to be deployed.

5. Concluding Remarks

This paper has shown how MRS behave cooperatively in underground terrains. The QL algorithm has been investigated for its potential quality of good communication, while the ACS algorithm was explored for good navigation properties. The strengths of the two algorithms have been harnessed and optimized to form a hybrid model QLACS which addressed the behavioural situation in MRS effectively.

The analytical solution of the new model was explained as shown in Algorithms 3 and 4 and Tables 7 and 10. The hybrid architecture for the model was also described in Figure 7. The experimental setup describing different navigation locations for two robots was shown in Figure 8.

The new hybrid model QLACS for cooperative behaviour of MRS in an underground terrain was proven able to find the optimal route and handle cooperation between two robots

Table 19: Route cost comparison for QL and QLACS.

Runs	QL (sec)		QLACS (sec)	
	Path cost for R1	Path cost for R2	Path cost for R1	Path cost for R2
1	20	20	10	10
2	32	19	10	10
3	28	14	10	10
4	27	27	10	10
5	39	30	10	10
Average	29.5	22	10	10

Table 20: Summary of scalability performance on QLACS.

Row numbers	Number of robots	Time (sec)	Number of states inspected			
			Robot 1 (R1)	Robot 2 (R2)	Robot 3 (R3)	Robot 4 (R4)
1	2	10.0006	4	3		
2	2	11.0007	4	3		
3	2	8.0005	4	3		
4	3	16.0009	1	3	3	
5	3	17.001	1	3	3	
6	3	12.0006	1	3	3	
7	4	11.0006	1	1	3	2
8	4	14.0006	1	1	3	2
9	4	10.006	1	1	3	2

effectively. The cooperative behavioural problem was narrowed down to two situations: navigational and cooperative behaviours. ACS was used to establish the shortest optimal route for two robots while the QL was used to achieve effective cooperation decisions. The results achieved after integrating the two algorithms showed a reduction in route costs and computation times, as shown in Figures 9, 10, 11, and 12. The comparative analysis between QL, ACS, and QLACS proved that the last-named is more robust and effective in achieving cooperative behaviour for MRS in the proposed domain.

The result of this work will be used for future simulation of MRS applications in hazardous environments. An indication of expansion of this research area has conspicuously surfaced in both real life implementations and model increase. The model used in this work can also be improved upon by increasing the state space.

Conflict of Interests

The authors declare that there is no conflict of interests regarding the publication of this paper.

Acknowledgments

The authors gratefully acknowledge the financial support of the Organisation for Women in Science for the Developing World (OWSDW), L'oreal-Unesco for Women in Science, and the resources made available by the University of Cape Town (UCT) and University of South Africa (UNISA), South Africa.

References

[1] J. Leitner, "Multi-robot cooperation in space: a survey," in *Proceedings of the Advanced Technologies for Enhanced Quality of Life (AT-EQUAL '09)*, vol. 37, pp. 144–151, IEEE Computer Society, Iaşi, Romania, July 2009.

[2] G. Dudek, M. R. M. Jenkin, E. Milios, and D. Wilkes, "A taxonomy for multi-agent robotics," *Autonomous Robots*, vol. 3, no. 4, pp. 375–397, 1996.

[3] A. Chella, G. Lo Re, I. Macaluso, M. Ortolani, and D. Peri, "Chapter 1. A networking framework for multi-robot coordination," in *Recent Advances in Multi Robot Systems*, A. Lazinica, Ed., pp. 1–14, 2008.

[4] Yinka-Banjo, C. O. ; Osunmakinde, and I. O. ; Bagula A, "Autonomous multi-robot behaviours for safety inspection under the constraints of underground mine Terrains," *Ubiquitous Computing and Communication Journal*, vol. 7, no. 5, pp. 1316–1328, 2012.

[5] S. Yarkan, S. Güzelgöz, H. Arslan, and R. Murphy, "Underground mine communications: a survey," *IEEE Communications Surveys & Tutorials*, vol. 11, no. 3, pp. 125–142, 2009.

[6] M. Peasgood, J. McPhee, and C. Clark, *Complete and Scalable Multi-Robot Planning in Tunnel Environments*, Digitalcommons, 2006.

[7] F. E. Schneider, D. Wildermuth, and M. Moors, "Methods and experiments for hazardous area activities using multi-robot system," in *Proceedings of the IEEE International Conference on Robotics and Automation*, pp. 3559–3564, May 2004.

[8] R. Zlot and A. Stentz, "Market-based multirobot coordination for complex tasks," *International Journal of Robotics Research*, vol. 25, no. 1, pp. 73–101, 2006.

[9] L. E. Parker, "Current research in multi-robot systems," in *Proceedings of the 7th International Symposium on Artificial Life and Robotics (ISAROB '03)*, vol. 7, pp. 1–15, 2003.

[10] C. Thorp and H. Durrant-Whyte, "Field robots," in *Proceedings of the 10th International Symposium on Robotics Research*, vol. 6, pp. 329–3240, 2003.

[11] S. R. Teleka, J. J. Green, S. Brink, J. Sheer, and K. Hlophe, "The automation of the "Making Safe" process in South African Hard-Rock underground mines," in *International Journal of Engineering and Advanced Technology (IJEAT '12)*, vol. 1, pp. 1–7, April 2012.

[12] L. Cragg and H. Hu, "Application of reinforcement learning to a mobile robot in reaching recharging station operation," in *Intelligent Production Machines and Systems*, pp. 357–363, 2005.

[13] R. A. C. Bianchi, C. H. C. Ribeiro, and A. H. R. Costa, "On the relation between ant colony optimization and heuristically accelerated reinforcement learning," in *Proceedings of the 1st International Workshop on Hybrid Control of Autonomous System*, pp. 49–55, 2009.

[14] E. Bonabeau, M. Dorigo, and G. Theraulaz, *Swarm Intelligence, from Natural to Artificial Systems*, Oxford University Press, Oxford, UK, 1999.

[15] Y. Liu and K. M. Passino, *Swarm Intelligence: Literature Overview*, Department of Electrical Engineering, University of Ohio, 2000.

[16] L. Li, A. Martinoli, and Y. S. Abu-Mostafa, "Emergent specialization in swarm systems," in *Intelligent Data Engineering and Automated Learning—IDEAL 2002*, vol. 2412 of *Lecture Notes in Computer Science*, pp. 261–266, 2002.

[17] B. Englot and F. Hover, "Multi-goal feasible path planning using ant colony optimization," in *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA '11)*, pp. 2255–2260, May 2011.

[18] Z. N. Naqiv, H. Matheru, and K. Chadha, "Review of ant colony optimization on vehice routing problems and introduction to estimated-based ACO," in *International Conference on Environment Science and Engineering (IPCBEE '11)*, vol. 8, pp. 161–166, 2011.

[19] M. Dorigo and T. Stützle, "The ant colony optimization metaheuristic: algorithms, applications and advances," Tech. Rep. IRIDIA/2000-32, 2000.

[20] R. Claes and T. Holvoe, "Cooperative ant colony optimization in traffic route calculations," in *Advances on Practical Applications of Agents and Multi-Agent Systems*, vol. 155 of *Advances in Intelligent and Soft Computing*, pp. 23–34, 2012.

Hindawi Publishing Corporation
Mathematical Problems in Engineering
Volume 2014, Article ID 234928, 11 pages
http://dx.doi.org/10.1155/2014/234928

Research Article

Heuristic Scheduling Algorithm Oriented Dynamic Tasks for Imaging Satellites

Maocai Wang,[1,2] Guangming Dai,[1] and Massimiliano Vasile[2]

[1] *School of Computer, China University of Geosciences, Wuhan 430074, China*
[2] *Department of Mechanical and Aerospace Engineering, University of Strathclyde, Glasgow G1 1XJ, UK*

Correspondence should be addressed to Maocai Wang; cugwmc@gmail.com

Received 4 March 2014; Revised 13 June 2014; Accepted 13 June 2014; Published 17 July 2014

Academic Editor: Yi Chen

Imaging satellite scheduling is an NP-hard problem with many complex constraints. This paper researches the scheduling problem for dynamic tasks oriented to some emergency cases. After the dynamic properties of satellite scheduling were analyzed, the optimization model is proposed in this paper. Based on the model, two heuristic algorithms are proposed to solve the problem. The first heuristic algorithm arranges new tasks by inserting or deleting them, then inserting them repeatedly according to the priority from low to high, which is named IDI algorithm. The second one called ISDR adopts four steps: insert directly, insert by shifting, insert by deleting, and reinsert the tasks deleted. Moreover, two heuristic factors, congestion degree of a time window and the overlapping degree of a task, are employed to improve the algorithm's performance. Finally, a case is given to test the algorithms. The results show that the IDI algorithm is better than ISDR from the running time point of view while ISDR algorithm with heuristic factors is more effective with regard to algorithm performance. Moreover, the results also show that our method has good performance for the larger size of the dynamic tasks in comparison with the other two methods.

1. Introduction

Because earth observation satellites (EOS) have many features such as wide coverage area and long duration and are without boundaries limitation, they have become an important means for exploring and researching earth resources and have been widely used in the fields such as land surveying, vegetation classification, crop growth trend assessment, natural disaster monitoring, and management of large-scale infrastructure projects as well as battlefield reconnaissance and ground military target identification.

Mission planning plays a key role in the whole process of earth observation. It directly affects the result of task completion. With the increase of the types of on-orbit satellites, as well as the increasingly complex requirements for observation data, how to optimize the scheduling of satellite resources to meet all types of observational requests has presented new challenges for satellite mission planning.

Satellite scheduling is to allocate the observation resources and executing time to a series of imaging tasks. In the recent years, many researchers have focused on different types of scheduling problems for EOS. For example, Parish [1] adopted the genetic algorithm to schedule as many supports as possible by a schedule builder program for 24-hour satellite range schedules. Wolfe and Sorensen [2] described the priority dispatch algorithm and the look ahead algorithm and then presented a novel genetic algorithm with two additional binary variables. Vasquez and Hao [3] formalized the daily photograph scheduling problem of EOS as a generalized version of the well-known knapsack model and developed a tabu search algorithm to solve the problem. Vasquez and Hao [4] also designed a partition-based approach to get the tight upper bounds for the daily photograph scheduling problem of EOS, and then a simplex-based linear programming relaxation and a relaxed knapsack approach were presented to solve the problem. In addition, Barbulescu et al. [5] compared simple heuristic method, local search method, and genetic algorithm and showed that the genetic algorithm had the best performance in the three algorithms for the larger and more difficult

problems. Frank et al. [6] used a constraint-based language to model the scheduling problem of EOS and put forward a stochastic greedy algorithm with heuristics search. Globus et al. [7, 8] compared some algorithms such as the genetic algorithm, hill climbing, simulated annealing, squeaky wheel optimization, and iterated sampling on two variants of a realistically-sized model of the EOS scheduling problem. Lin et al. [9] adopted Lagrangian relaxation and linear search techniques to generate a near-optimal, feasible schedule for satellite daily imaging scheduling problem. Also, Zufferey et al. [10] designed a tabu search and an adaptive memory algorithm to tackle the satellite range scheduling problem by employing the best ingredients of the graph coloring techniques to the problem. Wang [11] designed a dynamic insert tasks heuristic algorithm named as DITHA for imaging satellite scheduling problem, which comprised three basic processes: insert directly, insert by iteration method, and insert by replacing. Mansour and Dessouky [12] proposed a genetic algorithm to solve the SPOT5 selection problem by a new genome representation. Huang et al. [13] proposed a fast heuristic algorithm called FHTIA for electronic reconnaissance satellite dynamic scheduling problem, which comprised three basic operators: directly insert tasks algorithm, shift and insert tasks algorithm, and replace tasks algorithm. As well, Zhang et al. [14] presented an ant colony optimization approach with a guidance solution to avoid trapping in local optima for solving the satellite control resource scheduling problem. Wang et al. [15] presented a nonlinear model of the scheduling problem and developed a priority-based heuristic with conflict avoided to schedule EOS constellation. Marinelli et al. [16] developed a Lagrangian version of the Fix-and-Relax MIP heuristic to solve the large scale input variables for satellite scheduling problem. Sarkheyli et al. [17] modeled the scheduling problem as the graph coloring and proposed a new tabu search algorithm to solve resources scheduling in low earth orbit by a new move function. Wu et al. [18] presented a novel two-phase based scheduling method in task clustering phase and task scheduling phase, constructed an acyclic directed graph model, and utilized a hybrid ant colony optimization algorithm for satellite observation scheduling. Dishan et al. [19] constructed an integer programming model, designed the rolling horizon strategy, and proposed three dynamic scheduling algorithms with the rolling horizon strategy for the dynamic scheduling problem of EOS. Wu et al. [20] designed an acyclic directed graph model by constructing a subacyclic directed graph in each orbit for multisatellite scheduling problem oriented to emergency tasks and common tasks and then presented a hybrid ant colony optimization algorithm mixed with iteration local search by repairing the constraints to guarantee all solutions are satisfying the emergency task requirement. Wang et al. [21] proposed a rule-based heuristic algorithm by describing the dynamic factors such as the changes of task amount, task properties, and task constraints into a uniformed form as inserting new tasks for dynamic scheduling problem of earth observing satellites and designed two heuristic rules named as max-contention for retraction and min-occupation for insertion, respectively. Iacopino et al. [22] designed an innovative dynamic planning algorithm with a high-level of adaptability and responsiveness, and the algorithm was based on ant colony algorithm by applying stigmergy to find near-optimum solutions on a global level for the Disaster Monitoring Constellation.

This paper is organized as follows. After analyzing the dynamic properties of imaging satellite scheduling, the optimization model is proposed. Then, two heuristic algorithms and two heuristic factors are put forward to solve the problem. A description of the test example follows. After presenting the results of the tests, the ability and the applicability of the heuristic algorithms and the heuristic factors are analyzed to derive some conclusions.

2. Analysis for the Dynamic Properties

Imaging satellites scheduling problem may face many dynamic factors as follows.

(i) Insert a new task: according to the actual requirement, users may insert some new tasks when a scheduling scheme is being executed, especially some unexpected tasks, such as the monitoring of serious earthquakes and volcanic eruptions and forest fires.

(ii) Cancel some tasks arranged: due to the change of user requirements, some arranged tasks may be cancelled before they are scheduled.

(iii) Change the task properties: some tasks may change their properties because users change their requirements or because task attributes are not reasonable.

(iv) Uncertainty of weather conditions: some tasks arranged may not be completed or have poor quality due to the change of the weather conditions such as cloud cover.

In fact, except for the dynamic properties caused by the tasks, there are many dynamic uncertainties resulting from the resources. For example, some satellites may be out of use because of malfunction, or some satellites may change their attitudes or orbit parameters, which results in the dynamic changes from the resources. However, the change of the resources can be seen as a change of the task. If some tasks are arranged to be executed in advance by a satellite and the satellite is invalid later, these tasks can be seen as new insert tasks. Therefore, we research the dynamic scheduling only in the change of the tasks.

3. Model

This study mainly focuses on imaging satellite scheduling problem of point targets, which means each target is viewed as a point that can be observed by a single observation strip. In fact, the scheduling problem of area targets can be solved by dividing them into many point targets.

Different to the static scheduling, in addition to the tradition requirements, there are some new requirements for the dynamic scheduling. The common requirements for the dynamic scheduling are as follows.

As far as possible to maximize the observation income: the optimal observation income is the initial goal of scheduling scheme and is also the goal always pursued in the whole scheduling process. This goal is often difficult to achieve because of a variety of reasons. The complexity of the problem and the limitation of computing ability result in the difficulty to achieve the optimal scheduling income. So the decision makers often have to accept a satisfied solution, not the optimal solution.

As far as possible to complete all dynamic tasks: in general, the dynamic tasks are the emergency tasks, which need to be completed in time. Therefore, in dynamic scheduling, the first and foremost thing is to complete the dynamic tasks immediately when they are added to the task set.

As far as possible to minimize the difference between the new scheme and the old one: satellite application is a complicated process, and it needs special equipment and takes time to upload an instruction to the satellite. Once the scheduling scheme is adjusted on a large scale, there may be a series of influences on the decision. Although the better income may be gotten by rescheduling fully, it may lead to a large-scale adjustment for the scheduling scheme, which results in the phenomenon of "shock" for the scheduling scheme.

As far as possible to speed up the adjustment from the old scheme to the new one: in the actual process of imaging satellites scheduling, the requirement for the timeliness of dynamic adjustment is very high. If it takes too long time to adjust the scheme, maybe the new scheme generated just does not meet the requirement. The goal with the optimal income usually conflicts with the one with the fastest adjustment. Although the optimal income can be gained by rescheduling, the new scheme is usually difficult to meet the timeliness requirements.

3.1. Parameters. The scheduling problem in the paper is characterized by the following sets, input parameters, and decision variables.

Sets:

S: the set of satellites

T: the set of tasks

DT: the set of dynamic tasks

W: the set of the time windows

W^j: the set of the time windows observed by satellites j

W_i^j: the set of the time windows observed by satellite j for task i.

Parameters:

m: the number of the satellites

n: the number of all tasks

sn: the number of the tasks scheduled

dn: the number of the dynamic tasks

swn: the number of the time windows scheduled for static tasks

c_i: the observed income of task i

n_i^j: the number of the visible time windows of task i observed by satellite j

$w_{i,k}^j$: the kth time windows of task i observed by satellite j

$\mathrm{st}_{i,k}^j$: the start time the kth time windows observed by satellite j for task i

$\mathrm{et}_{i,k}^j$: the end time the kth time window observed by satellite j for task i

$\mathrm{ang}_{i,k}^j$: the swing angle when task i is observed by satellite j in its kth time window

pre_j: the setup time of satellite j for starting

$\mathrm{shut}d_j$: the stable time of satellite j for shutdown

lim_j: the maximum allowable running time of satellite j for each loop

$t\,\mathrm{min}_j$: the shortest running time for satellite j

$t\,\mathrm{max}_j$: the longest running time for satellite j.

Decision variables:

$x_{i,k}^j$: a Boolean variable that can be either 1 if the task i can be observed by the satellite j in the kth time window or 0 if not observed.

It is noted that there may be many visible time windows for every task observed by each satellite. We denote the number of the visible time windows of task i observed by satellite j as $|W_i^j|$.

3.2. Optimal Model. The optimization model proposed in the paper consisted of two optimization objectives. The first one is to complete dynamic tasks as many as possible. This objective is particularly important for the emergency scheduling. The second one is to maximize the observation income, which is the common objective for all imaging scheduling. In the model, three constraints are considered. They are as follows:

$$\begin{aligned} &\text{Max.} \sum_{i=1}^{dn}\sum_{j=1}^{m}\sum_{k=1}^{|W_i^j|} x_{i,k}^j, \\ &\text{Max.} \sum_{i=1}^{n}\sum_{j=1}^{m}\sum_{k=1}^{|W_i^j|} x_{i,k}^j c_i \end{aligned} \tag{1}$$

$$\text{St.} \quad x_{i,k}^{j} t\min_j \le x_{i,k}^{j}\left(\text{et}_{i,k}^{j} - \text{st}_{i,k}^{j}\right) \le x_{i,k}^{j} t\max_j, \quad \forall i \in T, \quad j \in S, \quad k \in W_i^j \tag{2}$$

$$\left[\text{st}_{i,k}^{j} - \text{et}_{i',k'}^{j} - \left(\text{pre}_j + \text{shut}d_j + \text{Tran}\left(\text{ang}_{i,k}^{j}, \text{ang}_{i',k'}^{j}\right)\right)\right] x_{i,k}^{j} x_{i',k'}^{j} \ge 0, \quad \forall i, i' \in T, \quad s \in S, \quad k \in W_i^j, \quad k' \in W_{i'}^j, \quad \text{st}_{i,k}^{j} > \text{et}_{i',k'}^{j} \tag{3}$$

$$\sum_{j=1}^{|S|} \sum_{k=1}^{|W_i^j|} x_{i,k}^{j} \le 1, \quad \forall i \in T, \tag{4}$$

where $\text{Tran}(\text{ang}_{i,k}^{j}, \text{ang}_{i',k'}^{j})$ means the switching time that the same satellite j adjusts its swing angle from $\text{ang}_{i,k}^{j}$ to $\text{ang}_{i',k'}^{j}$ for completing task i' after having completed task i when task i and task i' are adjacent.

Constraint (2) means that each available time window must be longer than the shortest running time of its observing satellite and shorter than the longest running time of the satellite; constraint (3) denotes that any two observation time windows must satisfy the switch time requirement; constraint (4) shows that each task is observed no more than one time by all satellites.

4. Scheduling Strategy

The satellite scheduling is a complex constraint optimal problem with many constraints. Moreover, the problem has been turned out to be an NP-hard problem. In fact, Barbulescu et al. [23] showed that the single resource range scheduling problem is equivalent to the problem of minimizing the number of tardy jobs on a single-machine with release dates, which enables the deterministic algorithms not suitable to solve the problem if the input size of the problem becomes larger. In fact, there are hundreds of input variables only in one day observation for the satellite scheduling problem. Therefore, in almost all satellite scheduling, nondeterministic algorithms are adopted to solve the problem.

In order to meet the high timeliness requirement of dynamic satellite scheduling, two heuristic algorithms and two heuristic factors are proposed in the paper. In the two heuristic algorithms, calculating the time windows between the satellites and the targets is a basic procedure. In our method, the calculating process includes three steps. Firstly, the geocentric angle is computed according to the position of the satellite and the location of the target. Then the direction of the swing angle towards left or right is determined based on the relation between the location of the target and the normal vector of the satellite orbital plane. Finally, the exact visible time windows between a given satellite and a certain target are computed by iterating based on dichotomy. The detailed process to calculate the time windows had been given in our previous paper [24].

4.1. IDI Algorithm. In the first heuristic algorithm proposed, new dynamic tasks are inserted directly or inserted by deleting repeatedly, which is named as IDI. All new tasks are firstly inserted to a waiting queue according to their priorities from low to high. Note that the new tasks are sorted according to their priorities from low to high, not from high to low. What is the purpose? It is to ensure that the low priority new tasks can have the chance to be arranged while the high priority tasks can be arranged by eliminating the low priority tasks. If the high priority tasks are firstly arranged, the low priority tasks will not have the chance to be arranged because the high priority tasks occupied the only time windows of the low priority tasks. Therefore, if the priorities are sorted from low to high, the low priority tasks are firstly arranged and then, when the high priority tasks are considered, they will be arranged if there is a free time window; even without a free time window, they can also be arranged by deleting the low priority tasks (Algorithm 1).

Then the first task in the queue will be inserted to a free time window directly if it does not conflict with any other task. If inserting directly fails, some tasks will be deleted for inserting the first task in the queue. The choice for the deleted task will depend on an evaluation function. The evaluation function is based on the conflict degree. Every task has a conflict degree δ, which is defined as the number of the tasks that conflict with the task. The evaluation function based on the conflict degree is defined as follows:

$$f_i = \frac{c_i}{(1+\delta_i)^2}, \tag{5}$$

where c_i denotes the observed income of task I and δ_i the conflict degree of task i. In the formula, the denominator $(1+\delta_i)^2$ is designed to avoid the conflict by enlarging the punishment.

The deleted task will be placed in the rear of the queue. Then all the process will repeat until the waiting queue is empty.

4.2. ISDR Algorithm. In the second heuristic algorithm proposed, a new task is scheduled by three ways: insert directly, insert by shifting, and insert by deleting, and then the deleted tasks are rearranged; here we name the algorithm as ISDR. If a new task does not conflict with other tasks, the task can be inserted directly. If inserting a new task directly fails, which means the new task conflicts with other tasks, the task can be inserted by shifting other tasks. If inserting directly and inserting by shifting do not succeed, some tasks will be deleted for inserting a new task. The deleted tasks will be added to a waiting queue. After all new tasks have been inserted, the deleted tasks in the waiting queue will be sorted according to their priority. Then the tasks will be scheduled in turn if there are intervals. By ISDR, the new tasks, especially emergency tasks, can quickly be inserted to the initial scheduling scheme (Algorithm 2).

Figure 1 is an example about inserting by shifting. As is shown in Figure 1, every task has two time windows. One is the visible time windows and the other one is the available time windows. As for task i, $\text{st}_{i,k}^{j}$ and $\text{et}_{i,k}^{j}$ are the start time and the end time of a visible time window observed by the

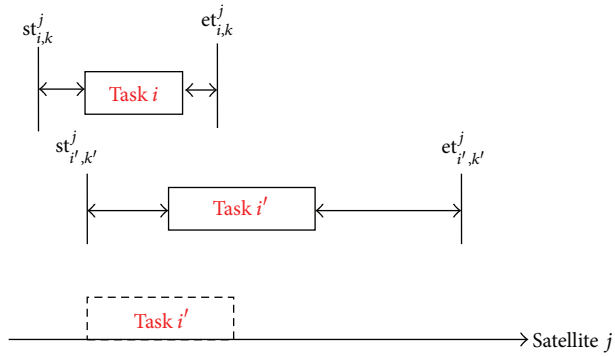


Figure 1: Insert by shifting.

(1) Set values for m, dn
(2) Sort dynamic task set DT according to the priorities from low to high
(3) for all $i \in [1, \ldots, \text{dn}]$ do
(4) for all $j \in [1, \ldots m]$ do
(5) Compute all visible time windows of task i observed by satellite j
(6) Compute the number n_i^j of the visible time windows of task i observed by satellite j
(7) end for
(8) end for
(9) while DT is not empty do
(10) Take the serial number i of the first task in the queue
(11) for all $j \in [1, \ldots, m]$ do
(12) for all $k \in [1, \ldots, n_i^j]$ do
(13) if the time windows$[st^j_{i,k}, et^j_{i,k}]$ does not conflict with others then
(14) Inserting the time windows$[st^j_{i,k}, et^j_{i,k}]$ directly
(15) goto Step 9.
(16) end if
(17) end for
(18) end for
(19) for all $j \in [1, \ldots, m]$ do
(20) for all $k \in [1, \ldots, n_i^j]$ do
(21) Calucate the conflict degree of $w^j_{i,k}$ with all time windows of the tasks scheduled
(22) end for
(23) end for
(24) Selecting the minimum conflict degree, take the serial number of the satellite as min j and the serial number of the time window as min k
(25) Deleting these scheduled tasks which time window conflict with the time window $w^{\min j}_{i,\min k}$
(26) Adding them to the rear of the queue
(27) Inserting the time window $w^{\min j}_{i,\min k}$ to the scheduled queue for task i
(28) end while

Algorithm 1: The algorithm of IDI.

satellite j. Similarly, $st^j_{i',k'}$ and $et^j_{i',k'}$ are the the start time and the end time of a visible time window of task i' observed by the same satellite j. The rectangle denotes the available time windows. Only if the actual time window of one task is arranged between the start time and the end time of its visible time windows and there is no conflict with other tasks, the actual time windows are the available time windows. As for task i', if the actual start time is located between the start time $st^j_{i',k'}$ of the visible time windows and the end time of the actual time windows of task i, there is a conflict between task i and task i' because the time windows overlap. So the conflict can be eliminated by shifting the actual start time of task i'.

4.3. Heuristic Factors. In multisatellite scheduling problem, each task may have many available time windows. Therefore, when there are a few available time windows for some task,

(1) Set values for m, dn
(2) Set wn = 0
(3) Sort dynamic task set DT according to the priorities from high to low
(4) for all $i \in [1, \ldots, \text{dn}]$ do
(5) for all $j \in [1, \ldots, m]$ do
(6) Compute all visible time windows of task i observed by satellite j
(7) Compute the number n_i^j of the visible time windows of task i observed by satellite j
(8) end for
(9) end for
(10) for all $i \in [1, \ldots, \text{dn}]$ do
(11) for all $j \in [1, \ldots, m]$ do
(12) for all $k \in \left[1, \ldots, n_i^j\right]$ do
(13) if $w_{i,k}^j$ not conflict with other time windows then
(14) Inserting $w_{i,k}^j$ directly
(15) goto Step 10.
(16) end if
(17) end for
(18) end for
(19) for all $j \in [1, \ldots, m]$ do
(20) for all $k \in \left[1, \ldots, n_i^j\right]$ do
(21) for all $t \in [1, \ldots, \text{sn}]$ do
if $w_{i,k}^j$ can inserted by shifting the time window of the tth task scheduled
(22) Insert $w_{i,k}^j$ by shifting the time window of the tth task scheduled
(23) goto Step 10.
(24) end if
(25) end for
(26) end for
(27) end for
(28) for all $j \in [1, \ldots, m]$ do
(29) for all $k \in [1, \ldots, n_i^j]$ do
(30) for all $t \in [1, \ldots, \text{sn}]$ do
(31) if the priority of the task t lower than the priority of task i then
(32) if $w_{i,k}^j$ can be inserted by deleting the time windows of task t then
(33) Insert $w_{i,k}^j$ by deleting the time windows of task t
(34) Add the task t to the waiting task list
(35) wn++
(36) goto Step 10.
(37) end if
(38) end if
(39) end for
(40) end for
(41) end for
(42) end for
(43) for all $i \in [1, \ldots, \text{wn}]$ do
(44) for all $j \in [1, \ldots, m]$ do
(45) for all $k \in [1, \ldots, n_{i,k}^j]$ do
(46) if $w_{i,k}^j$ not conflict with other tasks scheduled then
(47) Inserting $w_{i,k}^j$ directly
(48) goto *Step* 10
(49) end if
(50) end for
(51) end for
(52) end for

Algorithm 2: The algorithm of ISDR.

how to select the appropriate time window is a very important problem. If some reasonable heuristic factors can be adopted to find the best time window for every task, the performance of the scheduling algorithm can be improved remarkably.

In this paper, we proposed two heuristic factors, one is to select new task based on the congestion degree between the time windows. The other one is to minimize the overlapping degree.

The congestion degree of a time window for some task is defined as the number of the tasks which could not be scheduled because they are affected by the task when it is scheduled in the time windows. When a task is scheduled, the congestion degree of every time window for the task is computed. Then all of the time windows for the task are sorted depending on the congestion degree in the order from small to large. For every task, the actual time window is selected according to the congestion degree.

Given the satellite, the overlapping degree of a task in a certain time point t is defined as the number of the tasks, which can also start in the time point. If task i will be observed by the satellite in the time windows $[st_{i,k}^{j}, et_{i,k}^{j}]$, the overlapping degree of task i on every time point will be computed. For any point, if the task i can be arranged in the time point, we denote $pck_i = 1$. The overlapping degree of task i on the time point t denotes as $OD_i = \sum_{j \in DT, j \neq i} pck_j$. Once the overlapping degree is computed for every time point, the actual time windows for a task can be selected according to the overlapping degree. The lower the overlapping degree, the more the chances selected.

4.4. Comparing with Other Methods. Wang et al. [21] designed two heuristic rules named as max-contention for retraction and min-occupation for insertion, respectively. The basic idea of max-contention is, for each time window, to compute the contention of every task that can be observed in the window and as far as possible to give priority to the task that only has the time windows, no other windows. The rule can guarantee to arrange the tasks that have few time windows. However, the rule does not consider whether the tasks with low priority have the chance to be arranged. Moreover, it does also not consider the conflicts of the time window among the tasks, which maybe result in many other tasks not being arranged because the task is observed. As for the rule of min-occupation, when inserted, a task is arranged in the place with minimal occupation for the time windows. By computing the occupation for every time window, the rule considers the occupation for the time window between the task and other tasks. However, when inserting, the rule is only based on the occupation, and the priority of the tasks is not considered, which results in some emergency tasks with high priority not having any chance to be arranged because they occupied a large period in the time window.

Based on the independent arriving time and deadline of the imaging, Dishan et al. [19] proposed three algorithms named as AIS/RH, DIS/RH, and WIS/RH by combining with rolling horizon. The algorithms consider the arrival sequence, the deadline sequence, and the waiting sequence for new tasks. The main idea is first arrangement for early arrival or deadline tasks, which can ensure that the tasks with the earlier arrived time or deadline are observed. However, the algorithms do not consider the task priority and also do not involve the occupation or contention. Therefore, the algorithms cannot ensure the real time response for emergency tasks. Moreover, it is possible for some task to occupy a certain resource for a long time or result in many other tasks failure, which deteriorates the overall performance.

The algorithms proposed in this paper, both IDI algorithm and ISDR algorithm, are based on inserting directly or inserting by deleting. Moreover, two heuristic factors, the congestion degree or the overlapping degree, are designed to deal with the occupation and contention. As for the tasks to be inserted, they are sorted according to the priority from low to high, not from high to low, which ensures that the tasks with low priority have the chance to be observed. It depends on the conflict degree to decide which task will be deleted in order to insert a new task. The basic idea can ensure that the tasks deleted are the tasks that have most impacts on other tasks. Therefore, many tasks can be observed if the task is deleted, which improves the overall observation income.

5. Test

In the test, there are 145 static tasks selected from China cities. The initial scheduling scheme for the 145 static tasks is generated by genetic algorithm. Dynamic tasks are generated randomly in the longitude range from 75 to 120 and the altitude range from 20 to 45. In the test, five group dynamic tasks with different size are given. There are 11, 21, 61, 101, and 201 dynamic tasks in every group, respectively. In order to analyze the performance of the heuristic algorithms and the heuristic factors, seven group tests are designed depending on the different heuristic strategies, which are named as T1, T2, T3, T4, T5, T6, and T7. They denote the following means:

T1: IDI algorithm without any heuristic factor

T2: IDI algorithm with congestion degree

T3: IDI algorithm with congestion degree and the overlapping degree

T4: add the overlapping degree in the process inserting directly

T5: add the overlapping degree in the process inserting directly and inserting by shifting

T6: add the overlapping degree in the process inserting directly, inserting by shifting and by deleting

T7: ISDR without any heuristic factor.

The test results are shown from Table 1 to Table 5.

As is shown in Table 1, if there are only a few dynamic tasks, whether there is any heuristic factor is not important; that means the heuristic factor have a little effect for the small size dynamic tasks. The main reason is to have little conflicts when the size of the dynamic tasks is small, and all new tasks can be scheduled. For the running time, it is clear that the tests with some heuristic factors or some strategy of ISDR will be slower than the test without any heuristic

Table 1: The result for 11 dynamic tasks with 145 static tasks.

Test item	Number of dynamic tasks completed	Number of static tasks completed	Number of static tasks affected	Income	Time (s)
T1	11	145	0	2726	0.093
T2	11	145	0	2726	0.094
T3	11	145	0	2726	0.5
T4	11	145	0	2726	1.093
T5	11	145	0	2726	0.436
T6	11	145	0	2726	0.421
T7	11	145	0	2726	0.031

Table 2: The result for 21 dynamic tasks with 145 static tasks.

Test item	Number of dynamic tasks completed	Number of static tasks completed	Number of static tasks affected	Income	Time (s)
T1	21	145	2	3641	0.438
T2	21	144	2	3631	0.719
T3	21	144	2	3631	1.875
T4	21	145	2	3641	2.969
T5	21	145	2	3641	1.139
T6	21	145	2	3641	1.139
T7	20	144	3	3544	3.869

strategy because there are many loops and judgments in ISDR algorithm, which consumes a lot of time. It is interesting to note that T1 is slower than T7; maybe there are too many evaluations to decide which task should be deleted in T1. It can also be found that T4 is slower than T5 and T6.

As is shown in Table 2, if one or many strategies in ISDR are adopted, the running time is longer. However, the number of the tasks completed in T4, T5, and T6 are more than the number in T2 and T3, which shows that the result of ISDR is better than IDI when the size of the dynamic tasks increases. It should be noted that T1 has better result than T2 and T3 although there is no any heuristic factor or heuristic strategy in T1, which shows the heuristic factors do not work because there are almost no conflicts when the size of the new tasks is small. Clearly, comparing T7 with T4, T5 and T6, ISDR without any heuristic factor is poorer than ISDR with heuristics, which shows that heuristic factors play an important role in ISDR. It is also shown there is no obvious difference in each process heuristic factors are added. The results in T4, T5, and T6 are closer to the global optimum than those in T2 and T3, which shows ISDR improves the effect of the global optimal results.

As is shown in Table 3, from the running time point of view, IDI is rapider than ISDR. However, the optimal result is better in ISDR than in IDI by comparing T4, T5, and T6 with T1, T2, and T3. With the increase of the number of the dynamic tasks, IDI with heuristic factor is better than those without any heuristic factor because there are more conflicts by comparing with T1, T2, and T3. It also can be noted that ISDR without any heuristic factor is significantly poorer than ISDR with heuristic factors, which shows heuristic factors play an important role in ISDR again. This conclusion is consistent with the result in Table 2. Moreover, the heuristic factors added in the three processes of ISDR have little effect on the original scheduling scheme.

The results in Tables 4 and 5 also support the conclusion that IDI is better than ISDR from the running time point of view. Similarly, the other conclusions in Tables 1, 2, and 3 can be verified from Tables 4 and 5. That is to say, the optimal result is better in ISDR than in IDI. Moreover, with the increase of the number of the dynamic tasks, IDI with heuristic factor is better than those without any heuristic factor. Finally, ISDR without any heuristic factor is significantly poorer than ISDR with heuristic factors.

We also tested and compared our method with the methods proposed by Wang et al. [21] and Dishan et al. [19]. The test results are given in Tables 6 and 7.

In Table 6, there are 101 dynamic tasks. Obviously, the three methods have good performance for the test. It can also be seen that Qiu's method outperforms our method and Wang's method both in the performance and in the running time for the case with 101 dynamic tasks. It should be noted that all of 101 dynamic tasks and 145 static tasks are completed in Qiu's method, which shows that the method based on the idea of first arrangement for early arrival or deadline tasks is effective when the size of the new tasks is small because there are few conflicts between the time windows.

In Table 7, there are 201 dynamic tasks. With the increase of the number of the dynamic tasks, the performance of Qiu's method worsens rapidly, and many dynamic tasks could not be completed because of the conflicts. In the three methods, our method outperforms, and all dynamic tasks are completed in our method, which shows that our method is effective when the size of the new tasks is larger because

Table 3: The result for 61 dynamic tasks with 145 static tasks.

Test item	Number of dynamic tasks completed	Number of static tasks completed	Number of static tasks affected	Income	Time (s)
T1	61	141	9	6969	3.172
T2	61	142	7	6979	2.391
T3	61	142	7	6979	7.218
T4	61	145	8	7010	14.187
T5	61	145	6	7010	13.572
T6	61	145	6	7010	5.648
T7	58	138	17	6716	14.243

Table 4: The result for 101 dynamic tasks with 145 static tasks.

Test item	Number of dynamic tasks completed	Number of static tasks completed	Number of static tasks affected	Income	Time (s)
T1	101	134	17	10299	7.562
T2	101	136	16	10321	7.313
T3	101	137	15	10332	18.766
T4	101	144	17	10406	110.187
T5	101	142	22	10386	150.976
T6	101	143	32	10396	97.069
T7	93	125	33	9600	80.184

Table 5: The result for 201 dynamic tasks with 145 static tasks.

Test item	Number of dynamic tasks completed	Number of static tasks completed	Number of static tasks affected	Income	Time (s)
T1	201	105	53	18441	38.422
T2	201	112	51	18518	37.032
T3	201	109	50	18468	89.875
T4	198	123	50	18332	75.578
T5	201	120	57	18621	210.1
T6	201	131	67	18740	315.383
T7	168	103	57	15867	113.178

Table 6: Comparison for three methods (101 dynamic tasks).

Test item	Number of dynamic tasks completed	Number of static tasks completed	Number of static tasks affected	Income	Time (s)
We	101	143	32	10396	97.069
Wang	101	117	57	9833	107.596
Qiu	101	145	11	10402	77.391

Table 7: Comparison for three methods (201 dynamic tasks).

Test item	Number of dynamic tasks completed	Number of static tasks completed	Number of static tasks affected	Income	Time (s)
We	201	131	67	18740	315.383
Wang	200	107	84	13692	437.169
Qiu	163	139	41	15771	201.447

ISDR with the overlapping degree inserts new tasks as much as possible by inserting directly and shifting or deleting then inserting. It can be noted that Wang's method is also effective for the case of 201 dynamic tasks, which shows that the rules of the max-contention for retraction and the min-occupation for insertion play an important role.

As for the running time, as is shown in Tables 6 and 7, Qiu's method is fastest in the three methods, maybe because the rule of first arrangement for early arrival or deadline tasks is simple. Our method and Wang's method are slower because there are many judgments in the two methods.

6. Conclusion

In this paper, two heuristic algorithms are designed for the dynamic scheduling of imaging satellite. In one algorithm, new dynamic tasks are inserted directly or inserted by deleting firstly while there are four steps for scheduling the new dynamic tasks in another algorithm: insert directly, insert by shifting, insert by deleting, and reinsert for the tasks deleted. Moreover, two heuristic factors are also adopted to improve the algorithm's performance. The result shows IDI is better than ISDR from the running time point of view, and IDI is especially suitable for solving the problem with large-size dynamic tasks while the high timeliness is the principal consideration. On the contrary, ISDR with heuristic factors outperforms IDI with regard to algorithm performance, but it takes longer time.

In future work, we should be concerned with more real world problems by taking into account more complex constraints such as weather condition and lighting condition. In addition, ISDR algorithm needs to be improved for speeding up its running time.

Conflict of Interests

The authors declare that there is no conflict of interests regarding the publication of this paper.

Acknowledgments

Maocai Wang thanks the Special Financial Grant from China Postdoctoral Science Foundation (Grant no. 2012T50681), the General Financial Grant from China Postdoctoral Science Foundation (Grant no. 2011M501260), the Grant from the Fundamental Research Funds for the Central Universities, China University of Geosciences (Grant no. CUG120114), and the Grant from China Scholarship Council (Grant no. 201206415018). Guangming Dai thanks the Grant from Natural Science Foundation of China (Grant no. 60873107) and the 12th five-year pre-research Project of civil aerospace in China.

References

[1] D. A. Parish, *A genetic algorithm approach to automating satellite range scheduling [Ph.D. thesis]*, Air Force Institute of Technology Air University, 1994.

[2] W. J. Wolfe and S. E. Sorensen, "Three scheduling algorithms applied to the earth observing systems domain," *Management Science*, vol. 46, no. 1, pp. 148–168, 2000.

[3] M. Vasquez and J. K. Hao, "A "logic-constrained" knapsack formulation and a tabu algorithm for the daily photograph scheduling of an Earth observation satellite," *Computational Optimization and Applications. An International Journal*, vol. 20, no. 2, pp. 137–157, 2001.

[4] M. Vasquez and J. K. Hao, "Upper bounds for the SPOT 5 daily photograph scheduling problem," *Journal of Combinatorial Optimization*, vol. 7, no. 1, pp. 87–103, 2003.

[5] L. Barbulescu, A. Howe, J. P. Watson, and D. Whitley, "Satellite range scheduling: a comparison of genetic," in *Proceedings of the 7th International Conference on Parallel Problem Solving from Nature*, 2002.

[6] J. Frank, A. Jansson, R. Morris, and D. Smith, "Planning and scheduling for fleets of earth observing satellites," in *Proceedings of the 6th International Symposium on Artificial Intelligence, Robotics, Automation and Space*, pp. 18–22, Montreal, Canada, 2002.

[7] A. Globus, J. Crawford, J. Lohn, and A. Pryor, "A comparison of techniques for scheduling earth observing satellites," in *Proceedings of the 19th National Conference on Artificial Intelligence*, pp. 836–843, San Jose, Calif, USA, July 2004.

[8] A. Globus, J. Crawford, J. Lohn, and A. Pryor, "Scheduling earth observing satellite with evolutionary algorithms," in *Proceedings of International Conference on Space Mission Challenges for Information Technology*, Pasadena, Calif, USA, 2003.

[9] W. Lin, D. Liao, C. Liu, and Y. Lee, "Daily imaging scheduling of an earth observation satellite," *IEEE Transactions on Systems, Man, and Cybernetics A: Systems and Humans*, vol. 35, no. 2, pp. 213–223, 2005.

[10] N. Zufferey, P. Amstutz, and P. Giaccari, "Graph colouring approaches for a satellite range scheduling problem," *Journal of Scheduling*, vol. 11, no. 4, pp. 263–277, 2008.

[11] J. Wang, *Research on robust scheduling approach and its application of imaging satellite [thesis]*, China National University of Defense Technology, 2008.

[12] M. A. A. Mansour and M. M. Dessouky, "A genetic algorithm approach for solving the daily photograph selection problem of the SPOT5 satellite," *Computers and Industrial Engineering*, vol. 58, no. 3, pp. 509–520, 2010.

[13] X. Huang, L. Zhang, D. Qiu, and J. Zhu, "Fast heuristic algorithm for electronic reconnaissance satellite dynamic scheduling problem," *Journal of Chinese Computer Systems*, pp. 31–11, 2010.

[14] N. Zhang, Z. Feng, and L. Ke, "Guidance-solution based ant colony optimization for satellite control resource scheduling problem," *Applied Intelligence*, vol. 35, no. 3, pp. 436–444, 2011.

[15] P. Wang, G. Reinelt, P. Gao, and Y. Tan, "A model, a heuristic and a decision support system to solve the scheduling problem of an earth observing satellite constellation," *Computers and Industrial Engineering*, vol. 61, no. 2, pp. 322–335, 2011.

[16] F. Marinelli, S. Nocella, F. Rossi, and S. Smriglio, "A Lagrangian heuristic for satellite range scheduling with resource constraints," *Computers and Operations Research*, vol. 38, no. 11, pp. 1572–1583, 2011.

[17] A. Sarkheyli, A. Bagheri, B. Ghorbani-Vaghei, and R. Askari-Moghadam, "Using an effective tabu search in interactive resources scheduling problem for LEO satellites missions," *Aerospace Science and Technology*, vol. 29, no. 1, pp. 287–295, 2013.

[18] G. Wu, J. Liu, M. Ma, and D. Qiu, "A two-phase scheduling method with the consideration of task clustering for earth observing satellites," *Computers & Operations Research*, vol. 40, no. 7, pp. 1884–1894, 2013.

[19] Q. Dishan, H. Chuan, L. Jin, and M. Manhao, "A dynamic scheduling method of earth-observing satellites by employing rolling horizon strategy," *The Scientific World Journal*, vol. 2013, Article ID 304047, 11 pages, 2013.

[20] G. Wu, M. Ma, J. Zhu, and D. Qiu, "Multi-satellite observation integrated scheduling method oriented to emergency tasks and common tasks," *Journal of Systems Engineering and Electronics*, vol. 23, no. 5, pp. 723–733, 2012.

[21] J. Wang, J. Li, and Y. Tan, "Study on heuristic algorithm for dynamic scheduling problem of earth observing satellites," in *Proceedings of the 8th ACIS International Conference on Software Engineering, Artificial Intelligence, Networking, and Parallel/Distributed Computing (SNPD '07)*, pp. 9–14, Qingda, China, August 2007.

[22] C. Iacopino, P. Palmer, N. Policella, A. Donati, and A. Brewer, "Highly responsive MPS for dynamic EO scenarios," in *Proceedings of the 12th International Conference on Space Operations (SpaceOps '12)*, Stockholm, Sweden, June 2012.

[23] L. Barbulescu, J. P. Watson, L. D. Whitley, and A. E. Howe, "Scheduling space-ground communications for the air force satellite control network," *Journal of Scheduling*, vol. 7, no. 1, pp. 7–34, 2004.

[24] M. Wang, G. Chen, G. Dai, and Z. Song, "Research on scheduling algorithm for multi-satellite and point target task on swinging mode based on evolution algorithm," *Journal of Computer Applications*, vol. 33, no. 11, pp. 3144–3148, 2013.

Hindawi Publishing Corporation
Mathematical Problems in Engineering
Volume 2014, Article ID 185378, 10 pages
http://dx.doi.org/10.1155/2014/185378

Research Article

An Online Full-Body Motion Recognition Method Using Sparse and Deficient Signal Sequences

Chengyu Guo,[1] Jie Liu,[1] Xiaohai Fan,[1] Aihong Qin,[2] and Xiaohui Liang[1]

[1] *State Key Laboratory of Virtual Reality Technology and Systems, School of Computer Science, Beihang University, Beijing 100191, China*
[2] *Zhejiang University of Media and Communications, Zhejiang 310058, China*

Correspondence should be addressed to Xiaohui Liang; lxh@vrlab.buaa.edu.cn

Received 18 February 2014; Revised 20 May 2014; Accepted 20 May 2014; Published 10 July 2014

Academic Editor: Yi Chen

This paper presents a method to recognize continuous full-body human motion online by using sparse, low-cost sensors. The only input signals needed are linear accelerations without any rotation information, which are provided by four Wiimote sensors attached to the four human limbs. Based on the fused hidden Markov model (FHMM) and autoregressive process, a predictive fusion model (PFM) is put forward, which considers the different influences of the upper and lower limbs, establishes HMM for each part, and fuses them using a probabilistic fusion model. Then an autoregressive process is introduced in HMM to predict the gesture, which enables the model to deal with incomplete signal data. In order to reduce the number of alternatives in the online recognition process, a graph model is built that rejects parts of motion types based on the graph structure and previous recognition results. Finally, an online signal segmentation method based on semantics information and PFM is presented to finish the efficient recognition task. The results indicate that the method is robust with a high recognition rate of sparse and deficient signals and can be used in various interactive applications.

1. Introduction

In recent years, sensor-based human motion recognition has received a great deal of attention from researchers. Sensors have been adapted for large-scale movements to avoid shading and lighting problems. This has advantages over vision-based methods for special scenes and has allowed full-body motion recognition and sensor-based motion control to be applied in various fields, such as medical rehabilitation and interactive games.

Currently, motion control tasks are based on accurate and complete accelerations, as well as signals provided by other sensors. Unfortunately, these devices are expensive and not easily portable. In practice, sparse and low-cost sensors are more attractive, but they are usually accompanied by less information, more noise, and frequent signal deletion, making it difficult to acquire or reconstruct accurate position information and accordingly harder to achieve a proper online recognition result. Therefore, reconstructing human motion from signal features based on sparse and deficient signals has recently evoked much interest.

In light of the above problems, an online motion recognition method that adopts sparse, low-cost Wii Remote sensors (Wiimotes) as input devices is proposed. Because sparse, deficient linear accelerations cannot acquire accurate position information of human motion, a predictive fusion model, which combines fused hidden Markov model (HMM) with an autoregressive process, is presented. Considering the independence of each part of the human body, a hierarchical fusion structure of fused HMM is used to deal with human motion signals, which enhances the independent and cooperative expression of the classification model. The predictive capability of the model provided by the autoregressive process ensures robustness when dealing with noisy and deficient signals. Once the online recognition process is underway, a graph model that builds the transition between different motion types filters those motion types and reduces the recognition complexity of the predictive fusion model (PFM).

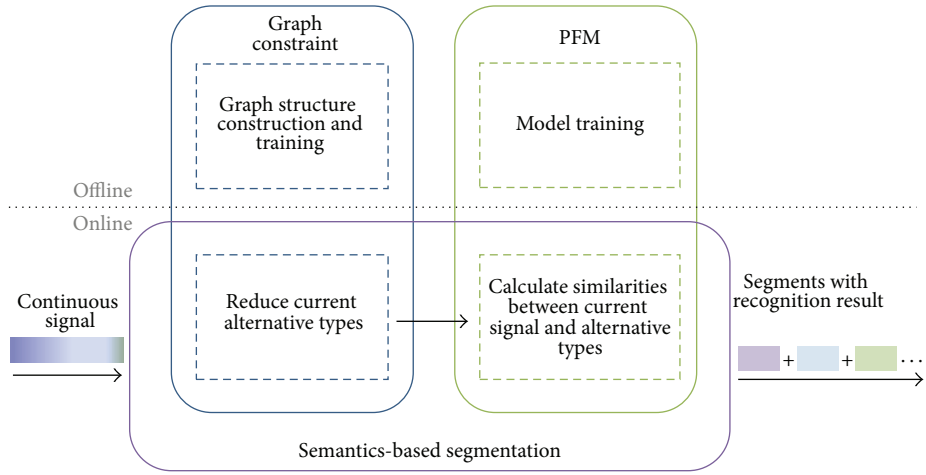


Figure 1: The structure of our method. The three main technologies include predictive fusion model (PFM), graph constraint, and semantics-based segmentation.

Moreover, a semantic-based automatic signal segmentation method is introduced to ensure the continuity of the online recognition processes.

Thus, based on sparse and deficient input signals, a human motion recognition PFM is presented that effectively supports sparse, low-cost sensors. The presented model is of a high accuracy rate and robust enough to handle insufficient and missing signals. An online motion recognition method is also proposed that does not require any position calibration. The method integrates PFM, action graph structure, and a semantic-based signal segmentation method to support user-driven virtual human motion in virtual scenes with continuous motions.

2. Related Works

As pattern recognition technologies develop, pattern recognition methods are increasingly used in the context of motion recognition. Typical methods, including self-organizing maps (SOMs), support vector machines (SVMs), and HMM approaches, can be adapted for motion recognition processes.

Methods for motion recognition vary depending on the input source. It has been shown that vision-based methods and sensor-based methods constitute two of the main research areas and are based on two types of input device, depending on the application. Poppe [1] presented a survey of vision-based human action recognition systems. Ning and Mokhtarian [2] used a shape to represent object contours extracted from each frame of a movie and constructed a tangent space based on the mean shape to approximate the linear space encompassing the datasets. Zhou et al. [3] and Min et al. [4] built a low-dimensional deformable model based on shape information from human motions in an image sequence to realize motion control. Lai et al. [5] proposed a local feature-based human motion analysis framework that extracted the features directly from local regions containing motion. Research has shown that the general idea of vision-based methods is to extract varied feature information from image sequences. In order to avoid the effect of light and shade and the inconvenience of vision-based methods when moving in a larger scene, sensor-based methods remain a hot topic in this field.

Recent work [6–9] which has described some basic methods for gesture recognition using accelerometers shows that sensor-based methods can be adapted for recognition tasks. Sun et al. [10] and Shiratori and Hodgins [11] used low-cost sensors to monitor daily physical activities. This method is practical but the finite types of simple activities limit recognition. Niu and Abdel-Mottaleb [12] considered the continuity of signals and provided a segmentation and recognition method based on HMM. Khan et al. [13] used a hierarchical scheme for human activity recognition. Tautges et al. [14] and Wong et al. [15] generated simple full-body animations controlled by sparse and accurate 3D accelerometers attached to the extremities of a human actor; this method is able to properly deal with accurate input to recover accurate human position information. In terms of both sparse and deficient signals, learning models are more effective than generative models. Early methods of the learning model define features analysis with HMM but require improvement in the robustness for deficient signals and the recognition rate.

The present research is motivated by the above studies. A probabilistic fusion model and autoregressive process in the hierarchical model of virtual human movement are proposed, which ensures that full-body motion information can be expressed relatively independently and deals with deficient input caused by sparse, inexpensive sensors. The recognition process ensures robustness, accuracy, and efficiency.

3. Method

3.1. Overview. In this paper, a recognition model PFM to deal with offline single motion segments is proposed first. Combined with graph constraint and online signal segmentation, the model can then be applied to online motion recognition. The method consists of three main key technologies, the structure of which can be found in Figure 1.

Predictive Fusion Model. Sparse and deficient inputs require more relevant information between each input signal sequence to keep local and global information. Therefore, HMMs of different part inputs were constructed, and a probabilistic model was used to fuse these HMMs together so as to enhance the model robustness. An autoregressive process is then introduced, which ensures that the unstable signals can be adjusted based on the past signals and training signals. The model can properly deal with offline motion recognition with sparse and deficient inputs.

Graph Constraint Construction. A graph structure based on the content of motion segments is constructed, limiting the choice for the following motion type based on the current motion content. The graph structure can filter part of motion type, reducing the complexity when dealing with a large motion database and improving the recognition accuracy as well.

Semantics-Based Signal Segmentation. Because input signals are continuous and may consist of multiple motion types, a method to separate the long continuous signal into segments was proposed. This method supports online motion recognition, the basis of the PFMs and graph constraint built offline.

3.2. Predictive Fusion Model. To build a robust learning model that can acquire feature information from sparse and deficient sequential input, HMM shows a high capability of dealing with time series. Here a predictive fusion model is presented based on the structure of HMM, which not only considers the sparse and deficient signal but also considers the features of human motion.

Consider two HMMs with observations $\mathbf{O}_1$ and $\mathbf{O}_2$, which indicate two groups of signal divided from all input sources, respectively. These input sources can be Wiimotes attached to different body parts in our experiments. For each motion type, a corresponding model is needed so as to value the similarity between the current input and the model, and the highest similarity probability determines the input type. Then, the problem can be defined as finding a solution to constructing the connections between the two HMMs so as to provide an optimal estimate for this similarity probability $p(\mathbf{O}_1, \mathbf{O}_2)$. To capture the statistical dependence between two observations $\mathbf{O}_1$ and $\mathbf{O}_2$, the maximum entropy principle is used:

$$p(\mathbf{O}_1, \mathbf{O}_2) = p(\mathbf{O}_1)\, p(\mathbf{O}_2)\, \frac{p(u, v)}{p(u)\, p(v)}, \tag{1}$$

where u and v are the respective transforms of $\mathbf{O}_1$ and $\mathbf{O}_2$ and absorb some dependence between $\mathbf{O}_1$ and $\mathbf{O}_2$. Here, $\langle u, v\rangle$ should be chosen from the two components of HMM, that is, the hidden state $\mathbf{S}$ and the observation $\mathbf{O}$.

Supported by the maximum mutual information criterion in [20], it is better to connect two HMMs by the hidden state sequence for one HMM and the observation sequence for the other one, rather than two hidden states for each one. The structure is shown in Figure 2. Thus, the transforms $\langle u, v\rangle$ can

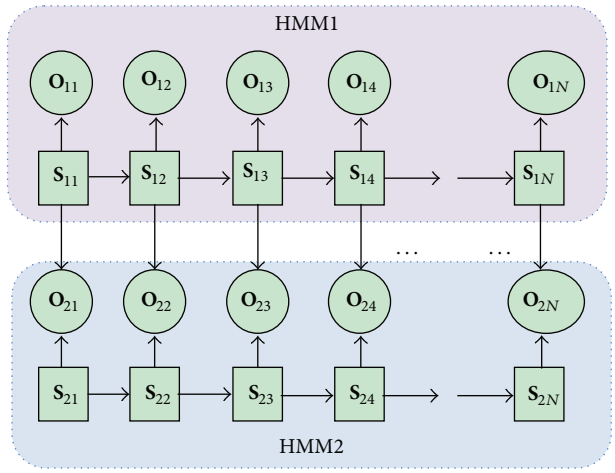


Figure 2: The structure of the model. A fusion relationship has been built between the hidden states of HMM1 and observations of HMM2.

be replaced by $\langle \mathbf{S}_1, \mathbf{O}_2\rangle$ or $\langle \mathbf{S}_2, \mathbf{O}_1\rangle$. The probability defined by (1) yields

$$\begin{aligned} p^1(\mathbf{O}_1, \mathbf{O}_2) &= p(\mathbf{O}_1)\, p(\mathbf{O}_2 \mid \mathbf{S}_1) \\ &= p(\mathbf{O}_1)\, p(\mathbf{O}_2)\, \frac{p(\mathbf{S}_1, \mathbf{O}_2)}{p(\mathbf{S}_1)\, p(\mathbf{O}_2)} \end{aligned} \tag{2}$$

or

$$p^2(\mathbf{O}_1, \mathbf{O}_2) = p(\mathbf{O}_2)\, p(\mathbf{O}_1 \mid \mathbf{S}_2), \tag{3}$$

where the structures defined by (2) and (3) are different. Equation (2) expresses the relationship between $\mathbf{S}_1$ and $\mathbf{O}_2$, indicating that the former HMM is more reliable than the latter one. The reliability of each HMM can be quantified as the weights for each part:

$$p(\mathbf{O}_1, \mathbf{O}_2) = \omega_1 p^1(\mathbf{O}_1, \mathbf{O}_2) + \omega_2 p^2(\mathbf{O}_1, \mathbf{O}_2), \tag{4}$$

where ω_1 and ω_2 represent the reliability of each body part motion. The values of ω_1 and ω_2 are determined by the selected types of actions. For general and daily activities, such as actions in our experiment, ω can be valued as 0.5, while for special occasion and activities, such as ping-pong, where the action focus is on the upper body part, ω can be valued as 0.8 and 0.2.

The observation $\mathbf{O}$ and state $\mathbf{S}$ can be unfolded as $\mathbf{O} = (\mathbf{o}_1, \ldots, \mathbf{o}_t)$, $\mathbf{S} = (\mathbf{s}_1, \ldots, \mathbf{s}_t)$, where t is the length of data sequence. The structure of this model is described in Figure 2.

Vary the basic parameters $\{\pi, \mathbf{A}, \mathbf{B}\}$ in HMM, where π stands for initial probability vector, $\mathbf{A}$ for state transition probability matrix, and $\mathbf{B}$ for observation probability vector, and the new parameters enhance the model's ability to deal with intermittent or noisy $\mathbf{O}$, where the hidden state $\mathbf{S}$ is taken into account in assuming $\mathbf{o}_t$, which can be written in the form of autoregressive process:

$$\mathbf{o}_t = e(\mathbf{s}_{t-1}, \mathbf{s}_t) + \sum_{i=1}^{p} \mathbf{c}_i(\mathbf{s}_{t-1}, \mathbf{s}_t)\, \mathbf{o}_{t-i} + \boldsymbol{\epsilon}_t, \tag{5}$$

where $\mathbf{e}_t$ is a parameter that preserves the descriptive power of the standard HMM when $\mathbf{c}_i = 0$ and $\boldsymbol{\epsilon}$ is residual error when calculating the observation $\mathbf{o}_t$. Since all current observations are affected by the current hidden state and past observations, parameter $\mathbf{B}$ of HMM can be modified as

$$\mathbf{B}^{12}(t) = \frac{1}{\sqrt{(2\pi)^D \left|\mathbf{K}\left(\mathbf{s}_{t-1}, \mathbf{s}_t\right)\right|}} \times \exp\left(-\frac{1}{2}\left(\boldsymbol{\epsilon}_t^{12}\right)^T \mathbf{K}\left(\mathbf{s}_{t-1}^1, \mathbf{s}_t^1\right)^{-1} \boldsymbol{\epsilon}_t^{12}\right), \tag{6}$$

where $\boldsymbol{\epsilon}$ can be calculated from (5).

The methods described above define the model parameters $\varphi = \{\pi_1, \mathbf{A}_1, \mathbf{e}_1, \mathbf{c}_1, \mathbf{K}_1, \pi_2, \mathbf{A}_2, \mathbf{e}_2, \mathbf{c}_2, \mathbf{K}_2, \mathbf{B}_{12}\}$, consisting of two predictive HMM parameters and the dependencies parameter $\mathbf{B}_{12}$. The training process can be summarized as follows.

(1) Calculate the parameters of two predictive HMMs with the expectation-maximization (EM) algorithm presented in [21] and Baum-Welch method in [22]. To maximize $P(\mathbf{O} \mid \boldsymbol{\varphi}), \mathbf{A}(\mathbf{s}_i, \mathbf{s}_j)\mathbf{B}(\mathbf{s}_i, \mathbf{s}_j)$ has to be maximized at each time t of the sequence, which can also be written as $\ln A(\mathbf{s}_i, \mathbf{s}_j) + \ln B(\mathbf{s}_i, \mathbf{s}_j)$. The terms that have to be maximized are

$$\sum_{t=1}^{T} \gamma_t\left(\mathbf{s}_i, \mathbf{s}_j\right)\left(\overbrace{\ln \mathbf{A}\left(\mathbf{s}_i, \mathbf{s}_j\right)}^{\text{transition term}} + \overbrace{\ln \mathbf{B}\left(\mathbf{s}_i, \mathbf{s}_j\right)}^{\text{observation term}}\right), \tag{7}$$

where $\gamma_t(\mathbf{s}_i, \mathbf{s}_j)$ is the probability of being in state $\mathbf{s}_i$ at time $t-1$ and in state $\mathbf{s}_j$ at time t in the Baum-Welch algorithm. To solve the terms in (7), the derivatives of the terms with respect to each variable $\mathbf{e}$ and $\mathbf{c}$ must be determined:

$$\begin{aligned} \sum_{t=1}^{T} \gamma_t \mathbf{o}_t - \sum_{t=1}^{T} \gamma_t \mathbf{e} - \mathbf{c} \sum_{t=1}^{T} \gamma_t \mathbf{O}_{\text{prior}} &= 0, \\ \sum_{t=1}^{T} \gamma_t \mathbf{o}_t \mathbf{O}_{\text{prior}}^T - \sum_{t=1}^{T} \gamma_t \mathbf{e} \mathbf{O}_{\text{prior}}^T - \mathbf{c} \sum_{t=1}^{T} \gamma_t \mathbf{O}_{\text{prior}} \mathbf{O}_{\text{prior}}^T &= 0, \end{aligned} \tag{8}$$

where $\mathbf{O}_{\text{prior}}$ indicates $\{\mathbf{o}_{t-1}, \mathbf{o}_{t-2}, \ldots, \mathbf{o}_{t-p}\}$. The parameters $\mathbf{e}(\mathbf{s}_i, \mathbf{s}_j)$ and $\mathbf{c}(\mathbf{s}_i, \mathbf{s}_j)$ can be calculated by solving (8). The covariance matrix $\mathbf{K}$ can then be calculated using the updated parameters $\mathbf{e}$ and $\mathbf{c}$:

$$\mathbf{K} = \frac{1}{N} \sum_{t=1}^{T} \gamma_t \boldsymbol{\epsilon}_t \boldsymbol{\epsilon}_t^T. \tag{9}$$

(2) Select one predictive HMM as the leading HMM and calculate the hidden state sequence for the leading HMM using the Viterbi algorithm. Then, determine the fusion parameters $\mathbf{B}_{12}$ or $\mathbf{B}_{21}$. If $\mathbf{O}$ is discrete, the following is obtained:

$$\mathbf{b}_i^{12}(j) = \frac{\sum_{t=1}^{T} \delta\left(\mathbf{o}_t^2, j\right) \delta\left(\mathbf{s}_t^1, i\right)}{\sum_{i=1}^{N} \delta\left(\mathbf{s}_t^1, i\right)}, \tag{10}$$

where N is the total hidden state number, j is the clustering number, and δ is the impulse function. When the parameter set φ of the model has been trained, the similarities $p(\mathbf{O}_1, \mathbf{S}_2)$ and $p(\mathbf{O}_2, \mathbf{S}_1)$ can be acquired by forward-backward algorithm, and the similarity $p(\mathbf{O}_1, \mathbf{O}_2)$ can be calculated by (2) or (3).

Then, how to use the model in the process of recognition will be shown. In training process, the input signal sequences $\mathbf{O}$ are four Wiimotes attached to all four human limbs, which are divided into two groups (upper and lower limbs). M models are trained for recognition use, where M indicates the total number of motion types. In the recognition process, the models trained for each motion type are used to compute the model's similarity to the input signal sequence. The solution to the similarity probability $p_M(\mathbf{O}_1, \mathbf{O}_2)$ can be calculated using the same forward algorithm as HMM. If the similarity to any motion exceeds a certain threshold, the sequence is classified as the motion type for which the similarity probability is the largest. The recognition result and similarity probability variation trend are shown in Figure 3. The results indicate that "waving hello" is the motion most similar to the input signal of the six types. Inspect the similarity probability of these models at each time, and it can be found that PFM had a higher classification capacity than the standard model because PFM can be determined timely at 20–40th frames. More experiments with larger databases will be described in Section 4.

3.3. Graph Constraint Construction. The model detailed above can properly identify the motion type from dozens of alternative ones. However, when the number of alternative motion types grows, it not only affects the accuracy rate of recognition but also increases the computation time due to the probability calculations required for each model. Therefore, a structured method was used to reduce the scale of alternative motion types in dealing with a large database.

When a user performs continuous and varied actions, it is noticed that certain action types cannot appear when the current action type has been determined, due to the coordination of human motion. This constraint can be used to guide selection of the following motion type based on the current determinate type.

The present graph model is motivated by the methods of Li et al. [23] and the motion graph of Kovar et al. [24] but different from the methods for different purposes and results. The model can be weighted or unweighted: the weighted one is a directed graph that contains the transition possibility detailing the compatibility and transitivity between two motion types. The node of the graph is of a single motion type, such as "walk" or "run." Before constructing the graph, a training process is necessary to obtain a more precise transition probability. Hundreds of long, continuous human motions are required, and the transition probability is calculated statistically by recording the frequency of motion transitions from one motion type to another. The unweighted graph has a similar structure to the weighted graph, but the transition probability only contains two values $\{0,1\}$. The structure of the graph is shown in Figure 4.

Once the recognition process is underway, the motion type is annotated immediately after recognizing the current

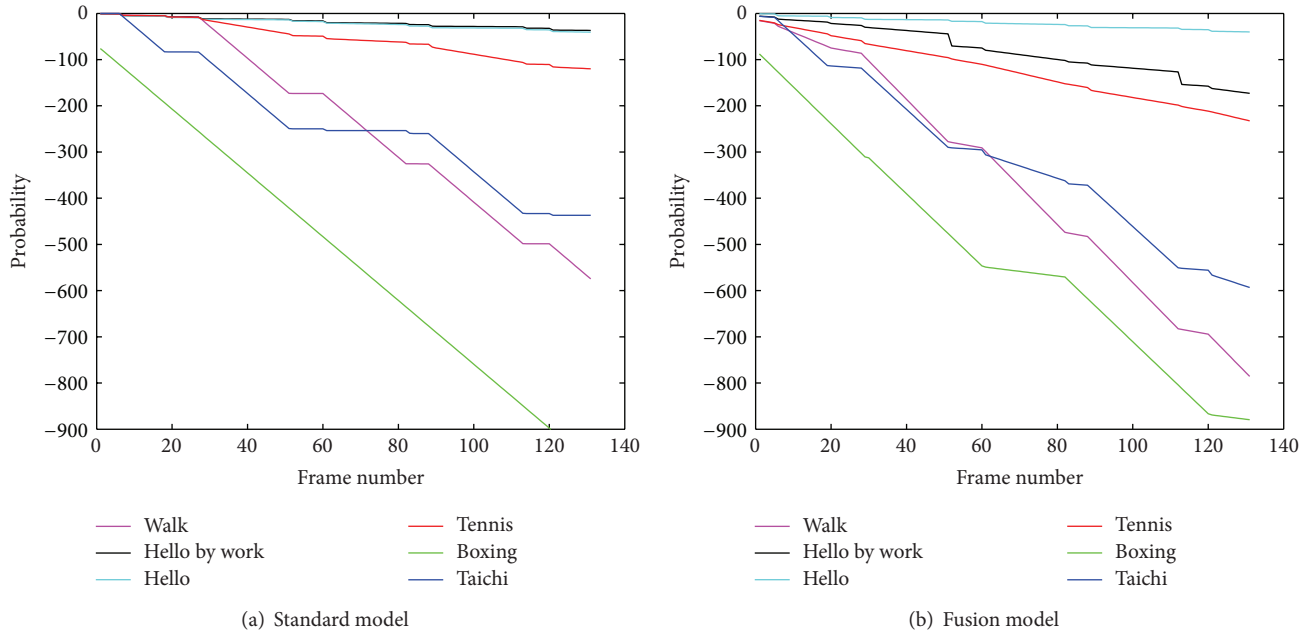
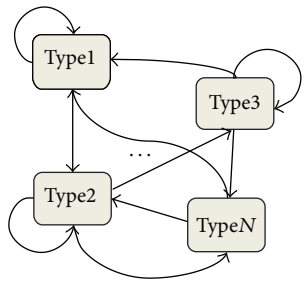


FIGURE 3: The accumulative results of similarity probability for HMM and PFM in a small scale database. The values on the y-axis indicate e^y, taking the logarithm of the probability, and the trend declines over time or frame number.

Graph	Type1	Type2	Type3	⋯	Type$N$
Type1	1	1	0	⋯	0
Type2	0	1	1	⋯	1
Type3	1	0	1	⋯	1
⋯	⋯	⋯	⋯	⋯	⋯
Type$N$	1	1	0	⋯	0

(a)



(b)

FIGURE 4: The constraint built by unweighted graphs. (a) The transition probability between two motion types. (b) The visual graph structure constructed from the table on the left. The transition probability [type x, type y] = 0 indicates that the type x is not permitted to follow the type y.

motion signal segment type i. For the unweighted graph, the nodes which are directed from node type i are selected, and the remaining motion types are excluded without calculating the similarity probability between the upcoming input signal and the current model. Only the models that correspond to selected nodes will calculate the similarity probability. For the weighted graph, the transition probability P_{ij} between two motion types i and j measures their similarities, as follows:

$$P_j(x) = \overbrace{f(P_{ij})}^{\text{transition term}} \overbrace{p_x(O_1)p(O_2)}^{\text{pfm term}}, \tag{11}$$

where x is one of the alternative motion types for the current signal segment and f is a scaling function that reduces the effect of P_{ij}, such as a logarithmic function.

3.4. Online Semantics-Based Signal Segmentation and Motion Recognition. For the online recognition process input signals which are always continuous and long need to be separated into short segments based on different motion types. In recent studies, such as the recursive least squares (RLS) method presented by [25] and the piecewise linear representation (PLR) method presented by [26], signal segmentation problems are always located at the break point in the signal energy curve, which may lead to oversegmentation or skipping smooth transition points. Therefore, signal segmentation based only on the signal shape is not comprehensive and requires consideration of the semantic information in the signal sequence.

In order to combine the semantic information with the segmentation process, the motion content needs to be parsed by a recognition model in the online signal segmentation. PFM is introduced into the process to acquire semantic information. With specific semantic information, it can be ensured that the segmented sequence is an intact and independent

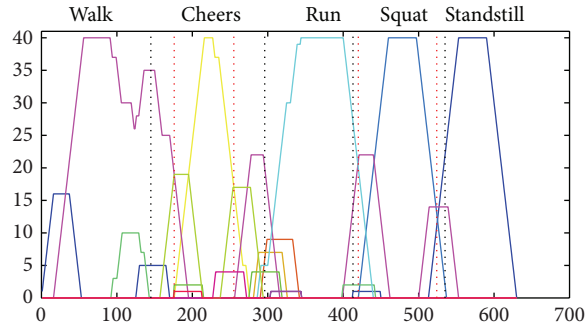


Figure 5: The accumulative process of similarity probability for HMM and PFM in a small scale database. The values on the y-axis indicate votes for each motion type at frame x. The red dotted line indicates segmentation points based on the method, and the black dotted line indicates breakpoints that the human actor expects.

motion type, which can greatly reduce the occurrence of oversegmentation. The method can be described as follows.

Let $\mathbf{O} = \{\mathbf{o}_1, \mathbf{o}_2, \ldots, \mathbf{o}_T\}$ be a long sequence of n-dimensional input acceleration signal vectors and let $v[T][N]$ be a two-dimensional integer vote array of time length T, where N is the number of all motion types. The array $v[t][k]$ indicates the number of votes of type k at frame t, which indicates the current motion type at t. On the online recognition stage, a sliding window of length M ($M << T$) scans the input sequence $\mathbf{O}$ from front to back with a step length of one frame. Each time the window is moved, the PFMs are programmed to recognize the signal segment in the current window M. For example, when the window moves to frame i, similarity probabilities are calculated by PFMs of alternative types with input of signals from $\mathbf{o}_i$ to $\mathbf{o}_{i+M}$. The vote array $\{v[i][k], v[i+1][k], \ldots v[i+M][k]\}$ will then be increased by 1, where k is the winner type in the present recognition. After the window sliding to frame p, N curves can be drawn based on $\{v[1 \cdots p][1], v[1 \cdots p][2], v[1 \cdots p][N]\}$ before frame p, which is shown in Figure 5. The intersection points shown in Figure 5 can be classified as alternative segment points, and recognition results can be acquired after finishing the segmentation.

To deal with transition signals and signals that do not belong to any alternative motion type, an appropriate threshold for each PFM should be set to filter out the redundant segment points during the PFM training process. The threshold is defined as the minimum normalized probability in the training dataset, and it rejects motion signals dissimilar to the training set.

The method presented above considers the semantic information of signal sequence and acquires the recognition result based on the PFMs trained offline. The recognition process is online, and results of which will be discussed in Section 4.

4. Results and Discussions

In this section, the functions of PFM, the effect of online recognition, and various applications of this technology will be described. As is presented in the last section, the input devices used in our experiment are sparse and low-cost (see Table 1). Devices with more information provided always result in higher price. Several general portable input devices are shown in the table, and sparser and cheaper devices are chosen to conduct our experiment. The signals analyzed here were the linear accelerations without any denoising or angular information, making it difficult to calculate accurate position information, as Table 2 shows.

The Wiimotes transmitted signals to a computer via a bluetooth interface that supports an 8–10 meters distance during an experiment. The sampling time in our experiment was 25 fps, which can be adjusted to accommodate a range of precisions. The training motion signal database has been preliminary constructed, which is clustered as 28 nodes in graph structure based on the content of the motion signal segments. Each node consists of 3-4 groups of motion segments with different variants, such as walking in different styles or kicking to different positions. Each type of motion signal is captured 5 times by 4 different actors. These hundreds of motion signals are well-organized for model training. In the experiment, thousands of independent action signals and hundreds of long continuous action signals are performed by testers in real time to get the result on recognition rate, robustness, and so forth.

4.1. Performance of Model. Before the experiment, we have tested several state-based methods, such as coupled HMM and structural HMM, as Pan et al. [20] presented. The result shows that fused HMM presents a better accuracy and robustness to the others when dealing with sparse and deficient motion signals. Therefore, in this section, the functions of our PFM will be shown and the accuracy and efficiency of the recognition process will be only compared with the performance of traditional Gaussian HMM and fused HMM when dealing with sparse and deficient input.

In our experiment, the recognition effect of different actors was validated by leave-one-out and k-fold cross validation methods, and the recognition rates of the PFM are shown in Figure 6, based on 40 alternative action types from the database we built. The HMM method yielded an average recognition rate of 42%, lower than the fused HMM and PFM recognition rates. The horizontal axis in Figure 6 represents the type of input signal sample and the vertical axis represents the types of corresponding models we built. While HMM is not robust when dealing with certain special motions, the PFM presents a more robust and accurate recognition result. In the HMM, without considering the motion of different body parts, the combined acceleration information led to confusion and presented a worse classification capability than for motions of similar variance. The fused HMM considered the structure of human motion and presented a higher classification capability than the standard model. The prediction capabilities of PFM were much better for these special inputs.

The proposed model can handle imperfect signals as well as deletion of input signals. The fewest number of sensors that can retain complete full-body motion information remains to be determined. Further experiments will be conducted to

Table 1: The details of general current portable input devices applied to motion control and recognition.

Sensor	Amount	Per price	Output information
Wii Remote	4	$39.99	3D linear accelerations, 2D rotation angle
Xsens' MTx [14, 16]	4 or more	$1500	Orientation, linear accelerations, angular velocity
MEMS sensors [17, 18]	8 for gait analysis	$250–8000	3D angular velocity, Orientation, etc
HD Hero [19]	16 or more	$250	Scene videos

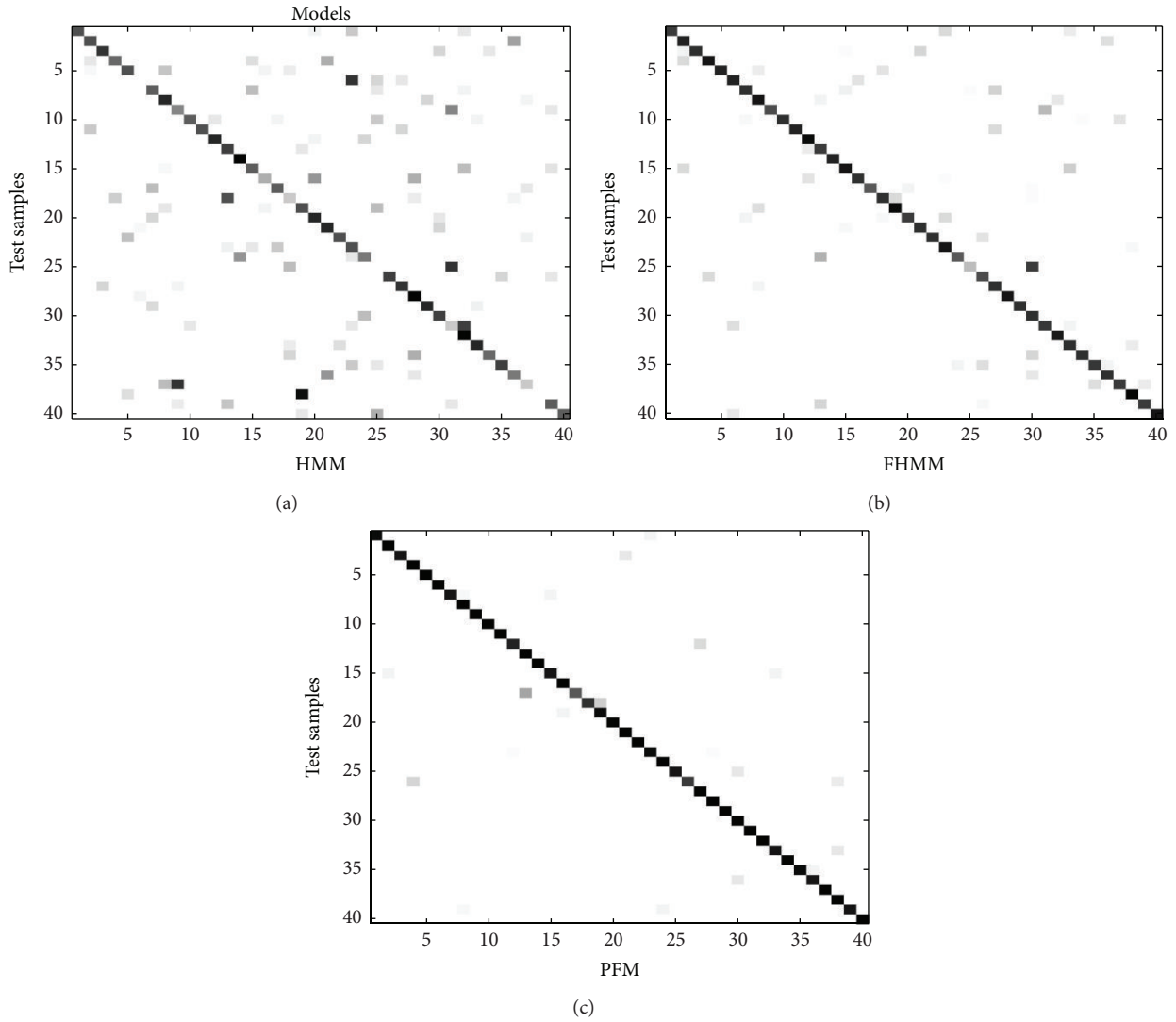


Figure 6: The recognition rate shown in the form of confusion matrices. With an increase in the recognition rate, the color of the matrix grid varies gradually from white to black.

Table 2: The offsets between actual motion data and position information, calculated by incomplete accelerations as the frame number increases.

Frame number	10	50	100	150	200
RMS value (cm)	9.55	22.13	49.68	77.86	122.46

Table 3: The PFM recognition rate for different actors with an increasing fraction of signal deletion.

Actor	Trained actor	New actor 1	New actor 2
Completed signals	0.97	0.92	0.91
One intermittent Wii	0.94	0.87	0.89
Two intermittent Wiis	0.85	0.84	0.85
One missing Wii	0.84	0.78	0.75

show the robustness and capabilities of dealing with deficient signals and to determine the requisite number of sensors in order to properly function in the motion recognition process. Table 3 shows the model's robustness with respect to deficient signals. In this experiment, different actors attached with reducing input devices are chosen. The actors here

Table 4: The accuracy rate of our segmentation method for different actors.

Actor	Actor 1	Actor 2	Actor 3	Actor 4	Actor 5
Desired points	48/50	50/50	47/50	50/50	49/50
Redundant points	8	10	5	12	7
Accuracy rate	0.86	0.83	0.9	0.8	0.87

include both trainees and newcomers. Since the action data of trainees are more standard and similar to trained motion data, the recognition rates for trainees are slightly higher than those for newcomers as the table shows. In the event that a short signal sequence from one sensor is lost, the recognition results remain unchanged from those derived from the complete signal sequence. For trained actors, the average recognition rate of HMM is 41% when two Wiis are intermittent and 73% for FHMM. This comparison shows that the classifying abilities of the PFM are greater than those of the two methods.

An analysis of unknown motions not included in the training datasets provides an estimate for the maximal probability of the motions most likely to be in the training datasets. Evaluation methods demonstrate the accuracy of the input signal relative to the recognition results.

4.2. Online Recognition. In an online recognition system, continuous signal processing is key for completing the task, and the results are essential for influencing and evaluating the recognition process. In our experiments, five actors were required to perform a continuous motion that included 51 motion segments used to test the segmentation accuracy rate. The accuracy rate of the segmentation experiment was evaluated by the number of desirable missing segmentation points and the number of undesirable or redundant segmentation points. Table 4 presents the segmentation results for these two measures. The desired segmentation points can be always located properly in our method, and the main factor that reduces the accuracy is the redundant segmentation points for our method. Unlike current segmentation methods of human motion signal sequence, that is, the RLS method and PLR method, whose abundant parameter and threshold groups are determined by repeated adjustments, our semantics-based method is more independent of parameters. For all this, a large number of desirable missing segmentation points for these two methods with an appropriate parameter group are always one of the main factors which may affect the accuracy rate of the segmentation. Besides, the delay of segmentation points and the accumulation of errors which always appear in these two methods can be effectively avoided in our method. Taking actor 3, for example, the desirable segmentation points are 41 for RLS and 36 for PLR, and the redundant segmentation points are 17 and 22, which is also more than semantics-based method. Figure 7 shows the final accuracy rate of these methods.

When dealing with large databases of alternative motion types, the difficulty in distinguishing features between different motion types becomes greater. The recognition capability of PFM is reduced substantially (see Table 5). Based on the graph structure we built, the alternative types of current segment recognition were fewer than the total alternative types, thereby preserving the online recognition accuracy rate rather efficiently. The high classification capability of the PFM model ensures that the results can be efficiently acquired at 30–50 frames of the signal input before the actor finishes the motion.

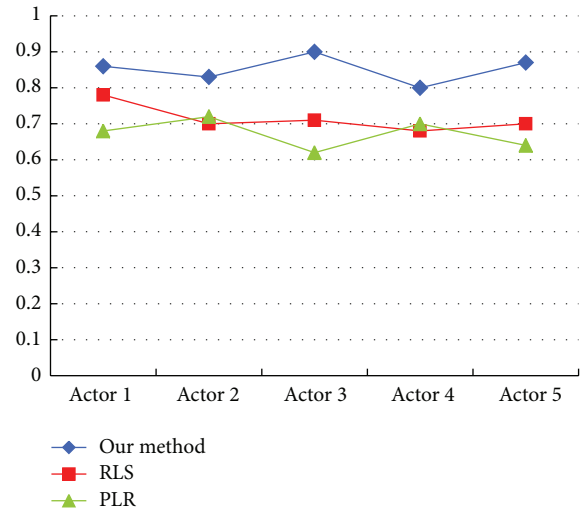


Figure 7: The segmentation accuracy rate of three methods. A long, continuous motion was performed by five actors.

4.3. Applications. The methods proposed here are applicable to a wide variety of applications, including behavioral teaching evaluations, interactive games in virtual environments, and activity validation systems in large-scale scenes.

A general application of the proposed recognition method includes driving the virtual human to generate computer animations or to simulate a virtual environment for user interactions. After the user performs the continuous motions the segmentation and recognitions are conducted efficiently, and the recognition results guide the searching process of the corresponding motion data in the database. The blending process in the motion graph technology guarantees continuity of the generated motion. Generative models, such as the Gaussian latent variable model presented by [27], can be properly embedded to synthesize more delicate and stylized motion in various applications.

In the context of educational applications, the present method can be used to evaluate activities, such as playing tennis, doing martial arts, or dancing. Students can act out motions while following a standard motion sequence that is presented in advance. The system can then evaluate the similarity of the mimicked sequence to the standard sequence.

Table 5: The recognition rate of PFM and PFM with graph constraint for trained actors with an increasing number of alternative motions.

Total alternative types	40	55	70	85	100
PFM	0.92	0.85	0.6	0.51	0.39
PFM with graph constraint	0.95	0.94	0.91	0.85	0.85

An evaluation system can be constructed by calculating the probability ratio between the input motion signal and the normative training data. The ratio provides an important evaluation criterion. The weights of the fusion model may be adjusted to standardize the motions of each appendage. Figure 8 shows an experiment based on the evaluation system described here. The proposed method was adapted to a set of complex motions associated with tennis, Tai chi, and boxing. The motions performed by the user were recognized and evaluated using our method.

Complex virtual environmental interactions constitute the main application focus of our method. Virtual environment games and special training regimens require environmental immersion and interactions with virtual objects. Our method, based on sparse, low-cost sensors, performed well in the context of these applications and can provide the user with an immersed experience.

Figure 8: The motion evaluation system. Left: an actor performs exercises with accelerometers attached to her four limbs. Right: the recognition result and the actor's performance grade.

5. Conclusion and Future Works

This paper presents a full-body motion recognition method based on sparse, low-cost accelerometers. In the online recognition process, a semantics-based signal segmentation method was adopted to acquire short motion segments, and a motion transition graph structure was constructed to reduce the amount of alternative motion types. To recognize the motion type accurately, a predictive fusion model was presented to efficiently distinguish between current motion types and alternative motion types. The models recognition capability is robust and accurate in dealing with unstable and deficient signals that provide little information for reconstructing position information. Results show that the method has a high recognition rate and can be adapted to specific input signals.

During experiments, it is found that the method had difficulty identifying the actors' orientation, as the input devices we used lack direction information for recovering whole motion information. In addition, a short pause in a continuous motion occasionally led to a redundant motion segment. In the future, in order to overcome these problems low-cost sensors will be integrated that will also provide direction information so that the input device can be more conveniently adapted to a specific interaction. The database of the motion signals and the motion data will also be expanded. Ultimately, the method will be applied to complicated scene interactions between users and the virtual environment.

Conflict of Interests

The authors declare that there is no conflict of interests regarding the publication of this paper.

Acknowledgments

This work was supported by the Natural Science Foundation of China (Grant no. 61170186) and the Zhejiang Leading Team of Science and Technology Innovation (2011R50019-06). The data used was obtained from HDM05 in [28] and CMU public database.

References

[1] R. Poppe, "A survey on vision-based human action recognition," *Image and Vision Computing*, vol. 28, no. 6, pp. 976–990, 2010.

[2] J. Ning and F. Mokhtarian, "Human motion recognition based on statistical shape analysis," in *Proceedings of the IEEE Conference on Advanced Video and Signal Based Surveillance (AVSS '05)*, pp. 4–9, September 2005.

[3] H. Zhou, L. Wang, and D. Suter, "Human motion recognition using gaussian Processes classification," in *Proceedings of the 19th International Conference on Pattern Recognition (ICPR '08)*, pp. 1–4, IEEE, December 2008.

[4] J. Min, Y. Chen, and J. Chai, "Interactive generation of human animation with deformable motion models," *ACM Transactions on Graphics*, vol. 29, no. 1, article 9, 2009.

[5] Y.-C. Lai, H. M. Liao, C.-C. Lin, J. R. Chen, and Y.-P. Luo, "A local feature-based human motion recognition framework," in *Proceedings of the IEEE International Symposium on Circuits and Systems (ISCAS '09)*, pp. 722–725, IEEE, Taipei, Taiwan, May 2009.

[6] J. Mlich, "Wiimote gesture recognition," in *Proceedings of the 15th Conference and Competition STUDENT EEICT*, vol. 4, pp. 344–349, Faculty of Electrical Engineering and Communication BUT, 2009.

[7] T. Schlömer, B. Poppinga, N. Henze, and S. Boll, "Gesture recognition with a Wii controller," in *Proceedings of the 2nd International Conference on Tangible and Embedded Interaction (TEI '08)*, pp. 11–14, Bonn, Germany, February 2008.

[8] P. Koch, W. Konen, and K. Hein, "Gesture recognition on few training data using slow feature analysis and parametric bootstrap," in *Proceedings of the International Joint Conference on Neural Networks (IJCNN '10)*, pp. 1–8, Barcelona, Spain, 2010.

[9] J. Pang and I. Singh, "Accelerometer based real-time remote detection and monitoring of hand motion," in *Proceedings of the World Congress on Engineering and Computer Science (WCECS '11)*, vol. 2 of *Lecture Notes in Engineering and Computer Science*, pp. 2078–2095, San Francisco, Calif, USA, 2011.

[10] L. Sun, D. Zhang, B. Li, B. Guo, and S. Li, "Activity recognition on an accelerometer embedded mobile phone with varying positions and orientations," in *Ubiquitous Intelligence and Computing*, pp. 548–562, Springer, Berlin, Germany, 2010.

[11] T. Shiratori and J. K. Hodgins, "Accelerometer-based user interfaces for the control of a physically simulated character," *ACM Transactions on Graphics*, vol. 27, no. 5, article 123, 2008.

[12] F. Niu and M. Abdel-Mottaleb, "HMM-based segmentation and recognition of human activities from video sequences," in *Proceedings of the International Conference on Multimedia and Expo (ICME '05)*, pp. 804–807, IEEE, July 2005.

[13] A. M. Khan, Y. Lee, S. Y. Lee, and T. Kim, "A triaxial accelerometer-based physical-activity recognition via augmented-signal features and a hierarchical recognizer," *IEEE Transactions on Information Technology in Biomedicine*, vol. 14, no. 5, pp. 1166–1172, 2010.

[14] J. Tautges, A. Zinke, B. Krüger et al., "Motion reconstruction using sparse accelerometer data," *ACM Transactions on Graphics*, vol. 30, no. 3, article 18, 2011.

[15] C. Wong, Z. Zhang, R. Kwasnicki, J. Liu, and G.-Z. Yang, "Motion reconstruction from sparse accelerometer data using PLSR," in *Proceedings of the 9th International Workshop on Wearable and Implantable Body Sensor Networks (BSN '12)*, pp. 178–183, May 2012.

[16] D. T. H. Lai, M. Hetchl, X. Wei, K. Ball, and P. Mclaughlin, "On the difference in swing arm kinematics between low handicap golfers and non-golfers using wireless inertial sensors," *Procedia Engineering*, vol. 13, pp. 219–225, 2011.

[17] B. Huyghe, P. Salvo, J. Doutreloigne, and J. Vanfleteren, "Feasibility study and performance analysis of a gyroless orientation tracker," *IEEE Transactions on Instrumentation and Measurement*, vol. 61, no. 8, pp. 2274–2282, 2012.

[18] C. Yang and Y. Hsu, "A review of accelerometry-based wearable motion detectors for physical activity monitoring," *Sensors*, vol. 10, no. 8, pp. 7772–7788, 2010.

[19] T. Shiratori, H. S. Park, L. Sigal, Y. Sheikh, and J. K. Hodginsy, "Motion capture from body-mounted cameras," *ACM Transactions on Graphics*, vol. 30, no. 4, article 31, 2011.

[20] H. Pan, S. E. Levinson, T. S. Huang, and Z.-P. Liang, "A fused hidden Markov model with application to bimodal speech processing," *IEEE Transactions on Signal Processing*, vol. 52, no. 3, pp. 573–581, 2004.

[21] P.-V. Borza, *Motion-based gesture recognition with an accelerometer [Ph.D. thesis]*, Babeş-Bolyai University, Cluj-Napoca, Romania, 2008.

[22] P. Kenny, M. Lennig, and P. Mermelstein, "Linear predictive HMM for vector-valued observations with applications to speech recognition," *IEEE Transactions on Acoustics, Speech, and Signal Processing*, vol. 38, no. 2, pp. 220–225, 1990.

[23] W. Li, Z. Zhang, and Z. Liu, "Expandable data-driven graphical modeling of human actions based on salient postures," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 18, no. 11, pp. 1499–1510, 2008.

[24] L. Kovar, M. Gleicher, and F. Pighin, "Motion graphs," *ACM Transactions on Graphics*, vol. 21, no. 3, pp. 473–482, 2002.

[25] K. Momen and G. R. Fernie, "Automatic detection of the onset of nursing activities using accelerometers and adaptive segmentation," *Technology and Health Care*, vol. 19, no. 5, pp. 319–329, 2011.

[26] E. Keogh, S. Chu, D. Hart, and M. Pazzani, "An online algorithm for segmenting time series," in *Proceeding of the 1st IEEE International Conference on Data Mining (ICDM '01)*, pp. 289–296, San Jose, Calif, USA, December 2001.

[27] S. Levine, J. M. Wang, A. Haraux, Z. Popović, and V. Koltun, "Continuous character control with low-dimensional embeddings," *ACM Transactions on Graphics*, vol. 31, no. 4, article 28, 2012.

[28] M. Müller, T. Röder, M. Clausen, B. Eberhardt, B. Krüger, and A. Weber, "Documentation mocap database hdm," Technical Report CG-2007-2, Universität Bonn, Bonn, Germany, 2007.

Hindawi Publishing Corporation
Mathematical Problems in Engineering
Volume 2014, Article ID 902039, 11 pages
http://dx.doi.org/10.1155/2014/902039

Research Article

Using Ignorance in 3D Scene Understanding

Bogdan Harasymowicz-Boggio and Barbara Siemiątkowska

Faculty of Mechatronics, Warsaw University of Technology, Ulica św. Andrzeja Boboli 8, 02-525 Warsaw, Poland

Correspondence should be addressed to Bogdan Harasymowicz-Boggio; mysticdrow@gmail.com

Received 7 March 2014; Revised 2 June 2014; Accepted 16 June 2014; Published 7 July 2014

Academic Editor: Dongbing Gu

Awareness of its own limitations is a fundamental feature of the human sight, which has been almost completely omitted in computer vision systems. In this paper we present a method of explicitly using information about perceptual limitations of a 3D vision system, such as occluded areas, limited field of view, loss of precision along with distance increase, and imperfect segmentation for a better understanding of the observed scene. The proposed mechanism integrates metric and semantic inference using Dempster-Shafer theory, which makes it possible to handle observations that have different degrees and kinds of uncertainty. The system has been implemented and tested in a real indoor environment, showing the benefits of the proposed approach.

1. Introduction

Recent years have brought a rapid advance in the area of mobile robotics and intelligent systems. At the same time, in many developed countries we observe processes such as population aging and increase of labor costs. For these reasons, since the modern society is already accustomed to the pervasiveness of advanced technology and intelligent devices, in the near future we can expect robots to become an indispensable part of our everyday life. Robots and artificial intelligence agents will be used by nonspecialists and thus will need to understand semantic concepts and instructions, which are natural in the human language. To accomplish this goal, a good understanding of the human environment by computers is an absolute necessity. Even though the field of computer vision has provided many useful tools for object recognition in the last decades, most of the current algorithms used for this task lack any kind of awareness of their own limitations and thus make it difficult to achieve a deeper understanding of the observed world. Besides robotics, the cognitive enhancement of computer vision would be of great value in such fields as automatic surveillance, augmented reality, medical imaging and expert systems, computer graphics, computer-aided geometric design, and industrial quality assurance.

Ignorance and uncertainty are natural, intuitive concepts used constantly in human cognition [1, 2]. The human mind is aware of the fact that most information regarding the surrounding world is not available to it at the moment. Such "knowledge about lack of knowledge" is used for reasoning with imaginative exploration of different possible scenarios and, if necessary, for planning actions that lead to the acquisition of additional relevant information. When processing visual stimulus, this kind of awareness allows us to quickly select the places where an object can be and where it certainly cannot.

Uncertainty also plays an important role in the process of human object recognition [3–5]. For example, we would not classify a lone stick resting against the wall as a broom. However, if it was coming out from behind a bucket, we would classify it as a possible broom, as we are aware that one of the ends is not visible and the item might be a broom. If an object is partially occluded, observed from far away, has a very small size, or the lighting conditions are bad, we are uncertain about it to some degree until we look closer. Such inference mechanisms are obvious and often performed subconsciously in our daily life. Nevertheless, the explicit use of uncertainty is rare in computer vision and robotics.

In our previous work [6] we have proposed an object recognition procedure consisting of 5 elastically designed

stages: features extraction, segmentation, hypothesis formulation, low-level inference, and high-level inference, where the usefulness of semantic relations between objects and part-based reasoning was demonstrated. However, the system was completely unaware of occlusions or any other form of perceptual limitations, which are explored in the present paper.

When data is obtained by an intelligent system in a real environment, we deal with two kinds of uncertainty: stochastic, which results from the fact that the system behaves in a random way, and epistemic, which results from lack of knowledge. In the field of robotics, when dealing with decision-making tasks under uncertain conditions, the classical probability theory is typically applied. However, it is not capable of capturing epistemic uncertainty, which in our opinion is a crucial aspect of scene understanding.

In order to integrate ignorance awareness into a holistic recognition and inference system, we rely on Dempster-Shafer theory (DST) and the generalized Markov random fields proposed in [6]. The possibility of applying DST for the simple task of isolated object classification has been previously suggested in [7, 8]. In order to use DST in general object recognition problems we must deal with several challenges such as ambiguous part and object belief integration, optimization of the belief function for a whole scene consisting of multiple known and unknown objects, and belief aggregation for large numbers of hypotheses which avoids saturation of the DST parameters.

In this paper we propose a novel, unified approach for integrating observations with knowledge about various kinds of perceptual limitations of a 3D object recognition system. To achieve this we apply Dempster-Shafer theory, which allows the explicit handling of evidence with different degrees of uncertainty. In order to improve recognition reliability we integrate this mechanism with the methods of contextual semantic inference presented in [6].

The rest of the paper is organized as follows: in Section 2 we discuss the state of the art. The background of Dempster-Shafer theory is presented in Section 3. Our method of object classification is described in Section 4. In Section 5 experimental results are presented. We draw conclusions in Section 6.

2. State of the Art

The recent development of 3D sensors and computer vision techniques has made it possible to describe the local properties of the closest environment of a mobile robot with reasonable accuracy [9, 10]. Knowledge about the robot environment is usually encoded in the form of a map. Most methods focus on the following two categories.

(i) Metric maps [11, 12] which represent some geometric features of the environment. The environment is represented as a grid of cells or as a feature-based map. This approach has two major problems: the size of the map grows with the size of the environment and the accuracy of the map largely depends on the size of a cell. Feature-based maps are attractive because of their compactness and they are very useful during the process of localization. However, the path-planning based on this kind of representation is time-consuming.

(ii) Topological maps [13] which represent relations between distinctive parts in the environment. This kind of maps has a graph structure, where nodes are used to denote some areas or places in the environment, and edges denote adjacency.

Researchers have been recently focused on topological and mixed (metric-topological) maps that are *semantic*—that is, maps that not only contain data concerning the geometry and relations between parts of the environment but also hold the meaning of various recognized, labeled elements of the observed world [14–16]. The semantic labels attached to the places and objects give information not only about their names but also about functionalities: the doorway indicates the transition between different rooms; that is, a meal can be prepared in the kitchen using ingredients from the fridge and so forth.

Over the last years many interesting object recognition techniques have been developed [14, 17, 18]. Most of these techniques are focused on analyzing various features of the objects present on the captured scene. Even though in many cases they perform well, they do not use any kind of knowledge about the vision system's constraints, which might be regarded as a drawback.

Some works explore the effects of possible occlusions in 2D vision systems for specific tasks such as path following [19–21] and people recognition [22]. The authors of [23] take advantage of the fact that occlusions occur in order to penalize impossible alignments of known objects in the scene. Probably the most advanced use of knowledge about occlusions in 2D computer vision is presented in [24], where the authors propose to create a mask of foreground "occluders," which influences the scores of feature-based object classification on the background and [25], where the authors use reoccurring occlusion patterns learned in the training process.

3. Dempster-Shafer Theory

Reasoning under uncertainty and using inexact knowledge is frequently necessary for real-world problems [20]. Observations, which are the main source of information about the environment of a mobile robot, are uncertain, meaning that some data can be missing, unreliable, or ambiguous. The knowledge representation can be also imprecise, inconsistent, or partial. If multiple inference rules are applicable, problems of contradictions between redundant rules and problems of missing rules are not uncommon. A formalism adequate to deal with real-world information should therefore allow us to express and quantify all these aspects and provide an algorithm of integration of data from multiple sources with different degrees of uncertainty.

In order to reduce the uncertainty usually probability theory is applied—either experimental (i.e., based on the

frequency of events) or subjective (based on expert assessment). The main problem with probability is its difficulty to deal with ignorance—probabilities must be assigned even if no information is available. It requires to go beyond the rules of current interest and consider all possibilities, so it is not capable of capturing epistemic uncertainty. Also, a probability given by a single number does not provide any means to distinguish ignorance and evidence conflicts. A proper measure of ignorance could be very useful to verify whether the available knowledge is sufficient to justify a decision of the system, as it would allow us to consider the additional option of postponing the decision until enough information is gathered. In a similar way, in order to draw conclusions based on various pieces of information, it would be useful to have a quantitative measure of evidence conflict. High degrees of conflict represent situations where decisions should be made more carefully.

Dempster-Shafer theory (DST) [26–28] of evidence is a formalism for uncertain reasoning which fits the mentioned requirements. This theory was designed in order to deal with uncertainty, ignorance, and conflicts. In comparison to probability theory instead of assigning probabilities to events (hypotheses), knowledge is encoded by assigning masses m to subsets of the set T (power set) of all possible events.

Consider

$$m : 2^T \longrightarrow [0, 1]. \tag{1}$$

The masses fulfill the following requirements:

$$\sum_{a \in 2^T} m(a) = 1, \qquad m(\phi) = 0, \tag{2}$$

where ϕ denotes the empty set. A belief measure is given by the function $\text{bel} : 2^T \rightarrow [0, 1]$:

$$\text{bel}(A) = \sum_{B \subseteq A, B \neq \phi} m(B). \tag{3}$$

A plausibility measure is given by the function $\text{pl} : 2^T \rightarrow [0, 1]$:

$$\text{pl}(A) = \sum_{B \cap A \neq \phi} m(B). \tag{4}$$

The complements of *belief* and *plausibility* are called *doubt* and *disbelief,* respectively. The difference pl(A) − bel(A) describes *uncertainty*. The process of data aggregation according to DST consists of the following steps.

(i) Degrees of belief for particular hypotheses are obtained based on facts, which are treated as information sources; for example, when a door handle is observed it supports the hypothesis that the robot is observing a door and denies the hypothesis that the robot is observing a fridge.

(ii) Dempster's rule is applied in order to combine degrees of belief and disbelief. Each fact (or source) can support any hypothesis with a belief degree between 0 and 1 and also deny any hypothesis with a disbelief degree (also called the belief degree for the negation of the hypothesis) between 0 and 1. Belief and disbelief in a hypothesis need not to sum to 1.

Dempster's rule of combination for two sources (1 and 2) is described as follows:

$$m_{1,2}(a) = \frac{\sum_{b \cap c = a} m_1(b) \cdot m_2(c)}{1 - \sum_{b \cap c = \emptyset} m_1(b) \cdot m_2(c)}, \tag{5}$$

where $a, b, c \subseteq T$. Based on (5) we obtain the following practical formulas:

$$\begin{aligned} m_{1,2}(h) &= \frac{m_1(h) \cdot m_2(h) + m_1(h) \cdot m_2(h_u) + m_1(h_u) \cdot m_2(h)}{1 - m_1(h) \cdot m_2(h_n) - m_1(h_n) \cdot m_2(h)}, \\ m_{1,2}(h_n) &= \frac{m_1(h_n) \cdot m_2(h_n) + m_1(h_n) \cdot m_2(h_u) + m_1(h_u) \cdot m_2(h_n)}{1 - m_1(h) \cdot m_2(h_n) - m_1(h_n) \cdot m_2(h)}, \\ m_{1,2}(h_u) &= 1 - m_{1,2}(h) - m_{1,2}(h_n), \end{aligned} \tag{6}$$

where h is the hypothesis, h_n is the negation of the hypothesis, h_u is the uncertainty of the hypothesis, and $m(*)$ is the belief (or supporting degree) function. Whereas the values $m(h)$ and $m(h_n)$ arise from evidence supporting or denying the hypothesis h and $m(h_u)$ represents the degree of uncertainty (i.e., lack of evidence altogether).

4. Inference under Uncertainty

As we have mentioned, we make use of our previous system [6]. In the first four, low-level processing stages, we integrate algorithms for extracting the uncertainty and occlusion data and we completely redesign the high-level inference stage. Before we present the details of our new approach, we briefly describe the function of the system stages.

In the first processing stage three geometric surface features are calculated for the whole point cloud. In our application we make use of an approximate, fast implementation of CAT features (convexity, anisotropy of convexity, and *theta* polar angle of the normal vectors). These are a set of intuitive, simple features that can be calculated quickly and used to describe the local properties of a surface. The convexity of a surface can be intuitively understood as a measure of how much the surface is convex nearby a selected point. A negative value of convexity would mean that the surface is concave. Anisotropy of convexity measures how much the convexity value varies depending on the direction for which it is measured. Therefore, the anisotropy of convexity for a spherical surface would be zero and for a cylindrical surface would be positive, though both surfaces have positive convexity. The *theta* angle represents the inclination of the surface relative to the gravity vector (measured with an

accelerometer). Even though this feature cannot always be applied, for most indoor objects, it is very useful.

Even though a variety of multidimensional descriptors have been developed to capture rich local surface properties [29–31], we have chosen the CAT features because of their high usefulness to distinguish simple shapes despite their low dimensionality. Due to the use of the *theta* inclination angle, these features are particularly useful in indoor environments, as they allow us to distinguish vertical and horizontal flat surfaces, which seem identical when analyzed with purely *intrinsic* shape descriptors. The benefits of applying a global z-axis reference have been shown by the authors of [32].

Let us discuss the proposed features in more detail. Consider a spherical coordinate system (R, φ, θ): radial distance R, azimuthal angle φ, and polar angle θ, beginning at a given point P which lies on the surface Π, with the axis $\theta = 0$ being the direction of the normal vector of the surface Π at the given point P. The surface Π is defined by

$$\theta = f(R, \varphi). \tag{7}$$

We define the surface convexity at P for a sphere radius R as

$$C_P(R) = \frac{1}{\pi R} \int_0^{\pi} \Psi(R, \varphi) + \Psi(R, \varphi + \pi)\, \mathrm{d}\varphi, \tag{8}$$

where

$$\Psi(R, \varphi) = \psi(R, \varphi) \cdot \delta(R, \varphi) \tag{9}$$

for $\psi(\varphi, R)$ being the θ angle of the vector $\mathbf{m}$, which is the projection of the vector normal to the Π surface at the point $[R, \varphi, f(R, \varphi)]$ onto the plane containing the axis $\theta = 0$ and the point $[R, \varphi, f(R, \varphi)]$. The δ function equals either 1 or -1. It is negative if and only if the vector $\mathbf{m}$ is pointed toward the line defined by $\theta = 0$.

Anisotropy of convexity is a measure of how much the convexity changes in different directions. For a point P on the surface Π (in a spherical coordinate system identical to the coordinate system used in (7)) anisotropy of convexity is defined for the sphere radius R by

$$\begin{aligned} A_P(R) = &\max_{\varphi \in [0,\pi)} \left[\Psi(R, \varphi) + \Psi(R, \varphi + \pi)\right] \\ &- \min_{\varphi \in [0,\pi)} \left[\Psi(R, \varphi) + \Psi(R, \varphi + \pi)\right]. \end{aligned} \tag{10}$$

These two features are both surface intrinsic properties. However, the objects commonly found in indoor environments have a well-defined base or set of possible base surfaces and thus lack degrees of freedom. This quality makes it possible to use the polar angle θ (i.e., inclination of the surface) as a valid feature of great discriminative power. This feature is invariant to rotation around any vertical axis.

After calculation, the CAT features are mapped into a 2D image, where the features, for convenience, are treated as 8-bit "colors" (ranging from 0 to 255)—this is done by assigning the features of each point to the corresponding pixel of the original depth image obtained from the sensor, from which the 3D coordinates were calculated. The features map is next smoothed using the mean-shift filtering algorithm described in [33] (implemented in the OpenCV library) with its spatial window radius set to 10 pixels and "color" window radius set to 20. After this operation, a canny edge detector is run and the resulting contours are connected using a simple morphological closing operation (with a square kernel of 3×3 pixels). The processed contours define a set of segments, which are further processed by rejecting the smallest ones (with an experimentally chosen area threshold of 100 pixels).

The next stage consists of formulating object parts hypotheses for the obtained segments. This is performed by matching feature histograms of the segments (i.e., the normalized histograms of CAT features for the pixels lying inside the segments) to the histograms of features calculated for "model" object parts (the model segments are obtained with the same segmentation algorithm, but using test scenes, where the objects are manually labeled). The histogram comparison is done by means of Pearson's correlation coefficient defined by

$$\mathrm{corr}(X, Y) = \frac{\mathrm{cov}(X, Y)}{\sigma_X \sigma_Y} = \frac{E\left[(X - \mu_X)(Y - \mu_Y)\right]}{\sigma_X \sigma_Y}. \tag{11}$$

If the correlation coefficient exceeds the required value ($\mathrm{corr}_{\min}$), an object part hypothesis is formulated. The correlation threshold is set individually for each part of the model object, depending on how unique the expected part features are. Each scene segment can have several hypotheses. A null hypothesis (hypothesis of unknown object) is additionally attached to each segment.

The low-level inference stage consists of iterative Monte Carlo matching of model point clouds provided as part of the system knowledge (described in [6]) to the scene segments whose feature histograms correlate with the known object parts. In this stage simple segments are combined into potential complex objects composed of multiple parts (these are denominated as object hypotheses or metahypotheses) by spatial model alignment. The successful alignment of several scene segments to the model usually does not provide sufficient evidence to terminate the recognition process. There are several reasons for this.

(1) Only successfully segmented patches of the scene are considered—the segmentation is imperfect, so many details are missing.

(2) Due to self and mutual occlusions, only a fraction of the full object is visible on the scene.

(3) The false-positive rate of recognition based on segment alignment is especially high for furniture regarded as a set of flat patches that can be aligned to several models.

The aim of the last processing stage (the high-level inference) is to find the best *scene theory*—the set of metahypotheses that best explains the observations taking into consideration additional, human-provided knowledge. This knowledge concerns the known objects (in our system we use boundaries for altitude above the ground level and parts importance weights) and their semantic relations with other

objects (three kinds of relations are currently used in the system, that is, *above*, *below,* and *beside*). This optimization is done using a DST belief degree (which is explained in this section) as an energy function.

4.1. Parts Present on the Scene. In the proposed system each part of a hypothetical object (whether attached to a particular scene segment or not) is regarded as an evidence source. For the found object parts (associated with an observed segment) the DST parameters are calculated with the equations presented below. For any given hypothesis let us define a few parameters:

$$q = 2 \cdot \frac{\operatorname{corr}\left(H_S, H_M\right)}{\left(1 + \operatorname{corr}_{\min}\right)}. \tag{12}$$

This parameter (q) contains the correlation of the segment feature histogram (H_S) with the model part histogram (H_M) relative to the minimal correlation required to formulate a hypothesis (i.e., the correlation threshold $\operatorname{corr}_{\min}$). q is truncated to fit into the range $[0; 1]$. Next, for any segment hypothesis of index j we define

$$s_j = q_j \cdot \frac{\sum_{i=0}^{N} q_i - q_j}{\sum_{i=0}^{N} q_i}, \tag{13}$$

where $\sum_{i=0}^{N} q_i$ is the sum of the q values for all the candidate hypotheses of the given segment (including j). The s value is equivalent to accumulated plausibility of the alternative part hypotheses of the scene segment. The normalization factor of this parameter is required (as will be shown through the rest of this section) to ensure that hypotheses of object parts that were actually found influence positively the belief in their combined object hypotheses (i.e., $m(h) > m(h_n)$).

Consider

$$u_{\text{area}} = 1 - \frac{\left|A_M - A_S\right|}{A_M} \tag{14}$$

This parameter (u_{area}) captures the uncertainty derived from deviations of the hypothetical part's surface area (A_S) compared to the model part's area (A_M). Such deviations are allowed to a certain point due to limited precision of the segmentation algorithm and partial occlusions.

Consider

$$u_{\text{dist}} = 1 - \frac{d}{d_{\max}}, \tag{15}$$

where d is the mean distance from the sensor to the segment points and $d_{\max}$ is the maximum sensor range. This value (u_{dist}) represents the uncertainty which comes from loss of precision and resolution for further objects. It should be noted that the linear precision loss model we used is a simplification and does not describe accurately the Kinect sensor. However, it reflects the optical linear relation of the size of the objects perceived as a 2D depth map to the sensor distance, which directly affects the segmentation algorithm. Both, u_{area} and u_{dist} are truncated to fit in the range $[0, 1]$.

We can now define the DST parameters of h—the hypothesis of presence of an object part—arising from a present (i.e., found) segment as follows:

$$\begin{aligned} m\left(h_u\right) &= 1 - u_{\text{area}} \cdot u_{\text{dist}}, \\ m\left(h\right) &= q\frac{1 - m\left(h_u\right)}{q + s}, \\ m\left(h_n\right) &= s\frac{1 - m\left(h_u\right)}{q + s} = 1 - m\left(h\right) - m\left(h_u\right). \end{aligned} \tag{16}$$

Therefore, the hypothesis uncertainty sums its *area* and *distance* components. The mass is based on the available evidence—the similarity of segment features to a model segment (q parameter). The disbelief is based on the accumulated evidence supporting the alternative possibilities (s parameter). As stated before, for found parts the equations are normalized to ensure $m(h) > m(h_n)$.

4.2. Absent Parts. The object parts which have not been found on the scene are also valuable sources of information. Their absence can provide strong evidence against a particular object hypothesis. However, the fact that a part has not been found does not necessarily lead to the conclusion that it is missing. The two main factors that can limit the part's visibility are occlusions and the camera frame boundaries. One of the novel features of the presented system is the ability to detect these factors and quantify them as uncertainty by using a basic form of spatial imagination described in this section.

To calculate the parts visibilities, the model object point cloud concerning a given hypothesis is aligned to the scene point cloud—this is performed using the transformation found during the low-level inference stage (for the reasons mentioned while discussing that stage, the segment-based alignment, is only a guess, not a sufficient recognition method). The model object is then projected into the flat camera coordinate system (knowing the intrinsic camera parameters). The depth of the projected model points is subtracted from the input scene depth at the corresponding points, calculating how many object points that are "closer" to the view point than the observed scene points. An example of such projection is presented in Figure 1.

Let us denote the depth map (2D pixel array) of the scene point cloud as a matrix $\mathbf{K} = [k_{ij}]$. The projection of the transformed model point cloud is a set of N points, consisting of x and y pixel coordinates and depths: $P = \{(x_i, y_i, d_i)\}$. The number of model points which should be visible can be calculated as

$$N_{\text{VMP}} = \sum_{i=0}^{N} T\left(k_{y_i x_i} - d_i - \varepsilon\right), \tag{17}$$

where ε is a small distance margin left in order to compensate alignment imperfections. T is a thresholding function, which assigns 1 to positive and 0 to nonpositive arguments. If some of the model object points fall out of the map boundaries, they are naturally excluded from this sum. The hypothetical

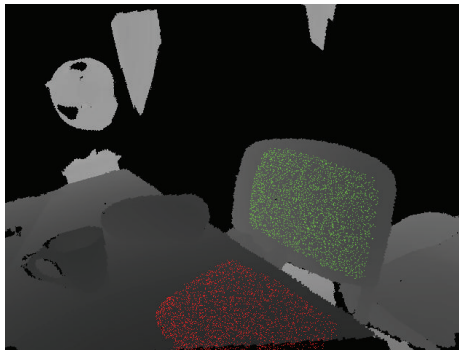


Figure 1: Assumed visible (green) and occluded (red) regions of a hypothetical object (a chair) obtained by spatial reasoning (output on the depth map).

visibility of a given part which has not been found is calculated simply as

$$v = \frac{N_{\mathrm{VMP}}}{N_{\mathrm{TMP}}}, \tag{18}$$

where N_{TMP} is the total number of model part points. We propose to calculate the DST parameters of h—the hypothesis of presence of an object part—arising from the fact that it was not found as

$$\begin{aligned} m(h) &= 0, \\ m(h_u) &= 2(1 - v), \\ m(h_n) &= 1 - m(h_u), \end{aligned} \tag{19}$$

where $m(h_u)$ is truncated to fit into the $[0, 1]$ range. Therefore, the disbelief of the hypothesis is based on the evidence pointing that the space where the missing part should be is empty (i.e., not occluded). The mass of the hypothesis is obviously zero. As we can see, a part whose visibility is less than 0.5 will have $m(h_u) = 1$ and thus no effect on the further calculations. This formula has been chosen in order to compensate for the imprecision of the segmentation algorithm near the borders between overlapping parts—if we expect most of a part to be occluded, it is likely that it has been omitted in the segmentation, so its absence is not regarded as relevant evidence. However, if a part is missing and it should be (based on the alignment results) in an area that is fully visible (i.e., empty area) with $v = 1$, it will strongly affect the hypothesis, as it will have $m(h_n) = 1$, meaning complete disbelief in the object part hypothesis.

4.3. Weights. The particular parts observations are not of equal importance—for example, a patch of flat surface can belong to many objects, whereas an ellipsoidal patch is uncommon and can provide strong evidence for the presence of a particular object such as a lamp. In the presented inference system, such part properties can be human-defined. In order to make it possible to include such kind of knowledge in the inference mechanism, we propose a "weighted" modification of the regular DST rule of combination. This

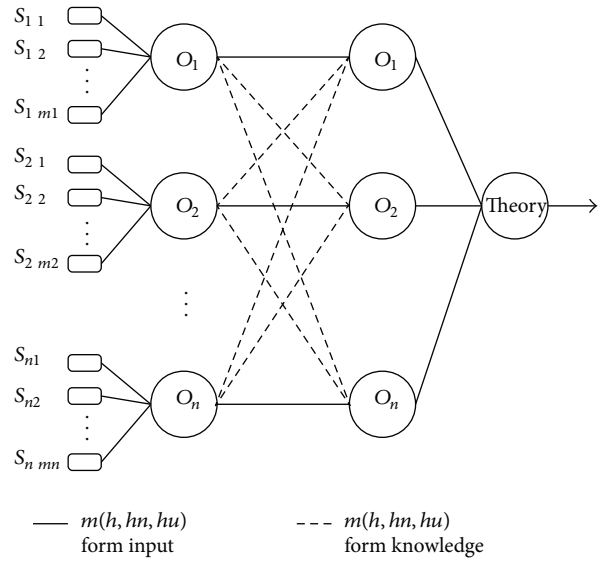


Figure 2: DST theory quality calculation scheme. Data flows from left (object parts DST parameters S_{nm}) to right (DST parameters for the whole scene theory).

modification consists of applying a weight w to the DST parameters of a hypothesis just before combination according to the following equations:

$$\begin{aligned} m_w(h) &= \frac{w \cdot m(h)}{w \cdot m(h) + w \cdot m(h_n) + m(h_u)}, \\ m_w(h_n) &= \frac{w \cdot m(h_n)}{w \cdot m(h) + w \cdot m(h_n) + m(h_u)}, \\ m_w(h_u) &= 1 - m_w(h) - m_w(h_n). \end{aligned} \tag{20}$$

The provided weights can be different for found parts (which we denominate as *positive weights*) and absent parts (which we call *negative weights*). The importance of this feature is best explained giving the example of a door. Detecting the vertical board is weaker evidence for the door presence than detecting the handle, but not detecting the board is stronger evidence against the door hypothesis than not detecting the handle.

4.4. Fusion. The high-level inference mechanism introduced in this paper uses a three-stage Dempster-Shafer fusion algorithm presented in Figure 2. The rectangles represent the DST parameters calculated for the parts of each object included in a scene theory (classification of all the scene segments) for both present and absent parts.

These parameters are integrated in the first stage of the fusion method (the metric stage) for each object in the weighted modification of Dempster's rule of combination described before. The resulting parameters $(m(h), m(h_n), m(h_u))$ for each object hypothesis are passed to the next stage (the semantic stage) and fused with a priori parameters that arise from the detected semantic relations with other objects present in the theory. These a priori parameters, fixed for each defined relation, are part

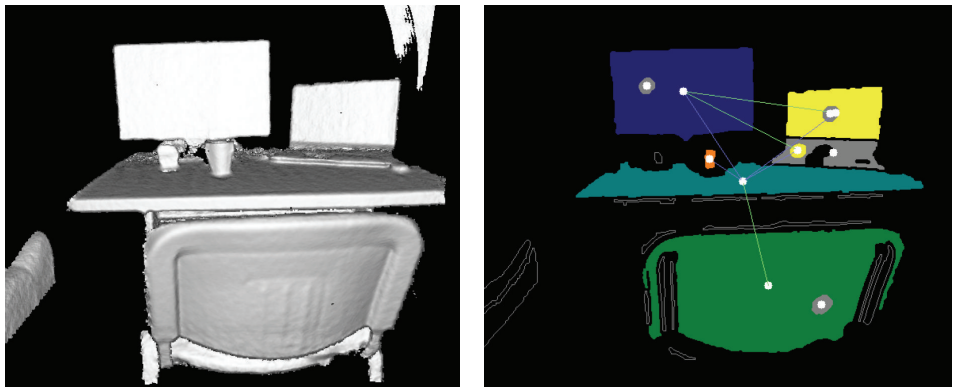


Figure 3: Recognition results for a simple scene (*monitor*—blue, *laptop*—yellow, *cup*—orange, *table*—turquoise, *chair*—green, and *unknown*—gray), including semantic relations: *beside*—green and *above*—blue.

of the human-provided knowledge of the system and can define either positive ($m(h) > 0, m(h_n) = 0$) or negative ($m(h) = 0, m(h_n) > 0$) relations. Obviously, for impossible or very unusual relations (such as *chair above monitor*) providing $m(h_n) = 1$ guarantees that the result of the fusion procedure will also be $m(h_n) = 1$, meaning certain rejection of such theory. Finally, the output parameters calculated for all the objects are combined together, resulting in a single set of DST parameters for the whole theory. Due to the tendency of Dempster's rule to converge to 1 for large quantities of plausible elements, the authors decided to apply a scaling factor for the $m(h)$ and $m(h_n)$ parameters resulting from the second processing stage (this factor has been experimentally set to 0.1). However, the scaling factor is applied only if the $m(h_n)$ parameter of a given object is less than 1, which prevents accepting impossible theories (contradictory to human-provided categorical rules).

Finding the best theory is a nontrivial optimization problem, which in our implementation is solved using a genetic algorithm, which aims to find the scene theory with the highest mass $m(h)$ (calculated according to the presented fusion mechanism) among all possible scene theories. The genetic algorithm initializes a population of theories (each classified scene segment is regarded as a gene), which then evolves for a fixed number of generations. Besides cross-over and mutation, each individual performs local optimization by single-gene substitutions (as described in [6]). The theory mass is used as the algorithm's fitness function. An example scene theory is shown in Figure 3.

4.5. Computational Complexity. In order to find the computational complexity for the worst-case scenario of the presented algorithms, we take into consideration the number of known object parts—N, which is roughly proportional to the number of known objects. The effort to calculate the proposed features and perform segmentation is proportional to number of points (P) present on the scene. Assuming that the scene resolution and size are fixed, for the first two processing stages, we get the order of complexity $O(P) = O(1)$.

The hypothesis formulation and low-level inference algorithms compare and align (for complex objects) each of the M scene segments to each model part. These two stages share the worst-case scenario order of complexity, which is $O(MN)$, as the number of maximum iterations of the Monte Carlo alignment method is fixed.

The last processing stage, as is currently implemented, has the highest worst-case order of complexity (even though in practice it has been observed to require about 1/3 of 4th stage's effort to complete). The used genetic algorithm requires just a $O(M)$-complex fitness function for the evaluation of each individual. However, the most complex part resides in the local optimization performed for each individual. If all the scene segments were assigned all the possible hypotheses and the local optimization went through all the possible scene theories, the order of complexity would be that of a brute-force search—that is, $O(M \cdot M^N) = O(M^{N+1})$. Even though there is no guarantee for a smaller complexity, the genetic algorithm proves much faster than a brute-force search. It would be possible to limit the maximum complexity by simplifying the local optimization to merely recalculating an individual's fitness for each single-gene substitution without further iterations. This would lead to a maximum order of complexity of $O(M^2N)$ but would not necessarily reduce the real amount of computation, as the genetic algorithm would probably need more generations to reach the best theory.

5. Experiments

The system described in the previous section has been implemented and tested using point clouds obtained using the Kinect Fusion algorithm [34] based on input from a Kinect sensor. The Kinect Fusion is a SLAM-based point cloud integration and filtering algorithm which has been used by the authors in order to enhance the input precision and to register reflective (e.g., a monitor) and dark surfaces which are very poorly captured in single Kinect frames (we used the open source version implemented in the Point Cloud Library by Anatoly Baskeheev). Scene views can be obtained with Kinect Fusion in real time, which requires only a few seconds of observation to achieve point cloud qualities significantly

superior to single frame point clouds. Such observations can be easily made by a mobile robot equipped with a Kinect-like sensor and a graphics processing unit (GPU).

Three sets of scenes (mostly related to an office environment) have been registered for the experiment as follows:

(1) a training set of 18 scenes from the office environment, which contain mainly fully visible objects observed from a short distance, used to extract the model point clouds and verify the human-provided parameters of the system (only some of these scenes were used to create the model objects);

(2) a test set of 14 realistic scenes of the office environment with known and unknown objects in different positions, captured from different distances; Occlusions occur but the environment presented in these scenes was relatively tidy and thus the set was denominated as the *easy* test set;

(3) a test set of 12 scenes with numerous, prevailing unknown objects and/or heavy occlusions. Some of these scenes are completely unrelated to the office environment; this set has been denominated as the *hard* test set; an example scene from this set is presented in Figure 4.

The system has been provided with 20 model views and 21 semantic relations, as well as the positive and negative part weights for complex objects. These parameters have been chosen using feedback from the test set only. In the experiment we used 9 known object classes: *chair, clock, cup, lamp, laptop, monitor, mouse, pencil case,* and *table*, of which 5 are treated as complex objects (i.e., composed of more than one part). The recognition performance of the system has been compared using the 26 test scenes for the following versions of the system:

(1) the system without any knowledge about its own perceptual limitations (i.e., limiting its belief on the formulated hypotheses based only on alternative hypotheses strength and provided part weights. However, hypotheses are not formulated for segments whose size is too big to possibly correspond to a given object part),

(2) the system aware of occlusions and limited field of view (i.e., using DST parameters for *absent* object parts as described in the previous section) but unaware of its limited precision,

(3) the system aware of its limited precision (i.e., using DST parameters for *present* object parts as described before) but not aware of occlusions and limited field of view,

(4) the *full* system aware of both kinds of limitations (using the DST parameters for all parts).

The performance was measured using 4 parameters:

(1) correct object recognitions relative to the number of known objects present on the scene;

(2) incorrect positive recognitions relative to the number of known scene objects (whether the incorrect decisions concern a known or unknown object);

(3) the $E1$ total error rate defined as

$$E1 = \frac{\mathrm{FN} + \mathrm{FP}}{\mathrm{NO}}, \tag{21}$$

where FN is the number of unrecognized known objects present on the scene (false negative), FP is the number of incorrect recognitions (false positive), and NO is the number of visible known objects;

(4) the $E2$ weighted error rate, calculated similarly to the $E1$ parameter, but using a weighting factor

$$E2 = \frac{0.2 \cdot \mathrm{FN} + \mathrm{FP}}{\mathrm{NO}}. \tag{22}$$

This error rate takes into consideration the fact that usually the cost of an incorrect positive recognition (false positive) for a mobile robot is higher than the cost of missing a known object (false negative), as the first case is more likely to cause counterproductive or even dangerous actions.

Figure 5 summarizes the experiment results. As we can see, the use of ignorance awareness slightly decreased the average correct recognition rate. However, it dramatically reduced the incorrect recognition rates as well as the error rates for both test sets, thus improving the system reliability. The results suggest that the use of *present* part uncertainty has higher impact on the system performance than the use of *absent* part uncertainty. However, combining both kinds of limitation awareness leads to the best overall results. As expected, the advantage of processing both kinds of ignorance was greater for the *hard* test set. For the *easy* test set, even though the incorrect recognition rate was lowest for the *full* system, the $E1$ total error rate was even slightly higher for the *full* system compared to the version with only *present* part uncertainty. The cause of this is the fewer numbers of unknown objects with misleading similarities to known object parts in this set. However, when taking into consideration the unequal cost of both kinds of errors (i.e., false positive and false negative), the *full* system performed best for this set as well.

The overall, relative reduction of error rates for the *full* system was of 12.5% and 35.9% compared to the version without *absent* part uncertainty: 58.8% and 79.0% compared to the version without *present* part uncertainty and 68.4% and 85.3% compared to the version without any kind of uncertainty for the $E1$ and $E2$ error rates, respectively.

The presented algorithms have been implemented in C++ CPU code, only partially exploiting multicore possibilities (at the low-level inference processing stage). The full processing time for a single scene depends on its complexity and ranges up to one minute.

6. Conclusions

In this work we have proposed a method to obtain and use information about the perceptual limitations of a 3D object

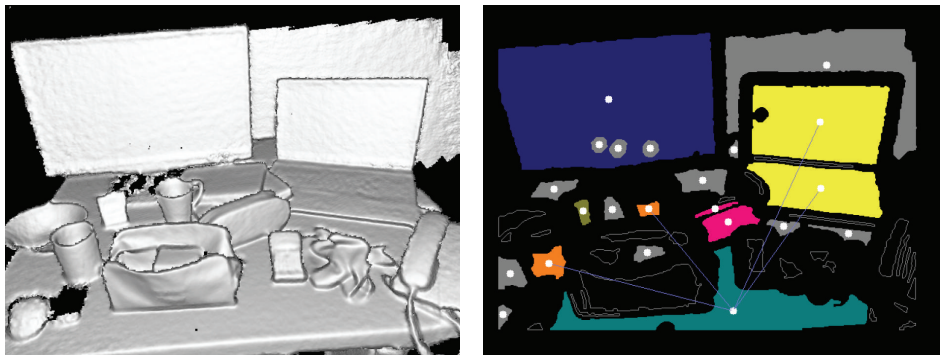


Figure 4: One of the test point clouds enhanced with kinect fusion, along with the recognition results (*monitor*—blue, *laptop*—yellow, *cup*—orange, *table*—turquoise, *clock*—olive, *pencil case*—pink, and *unknown*—gray).

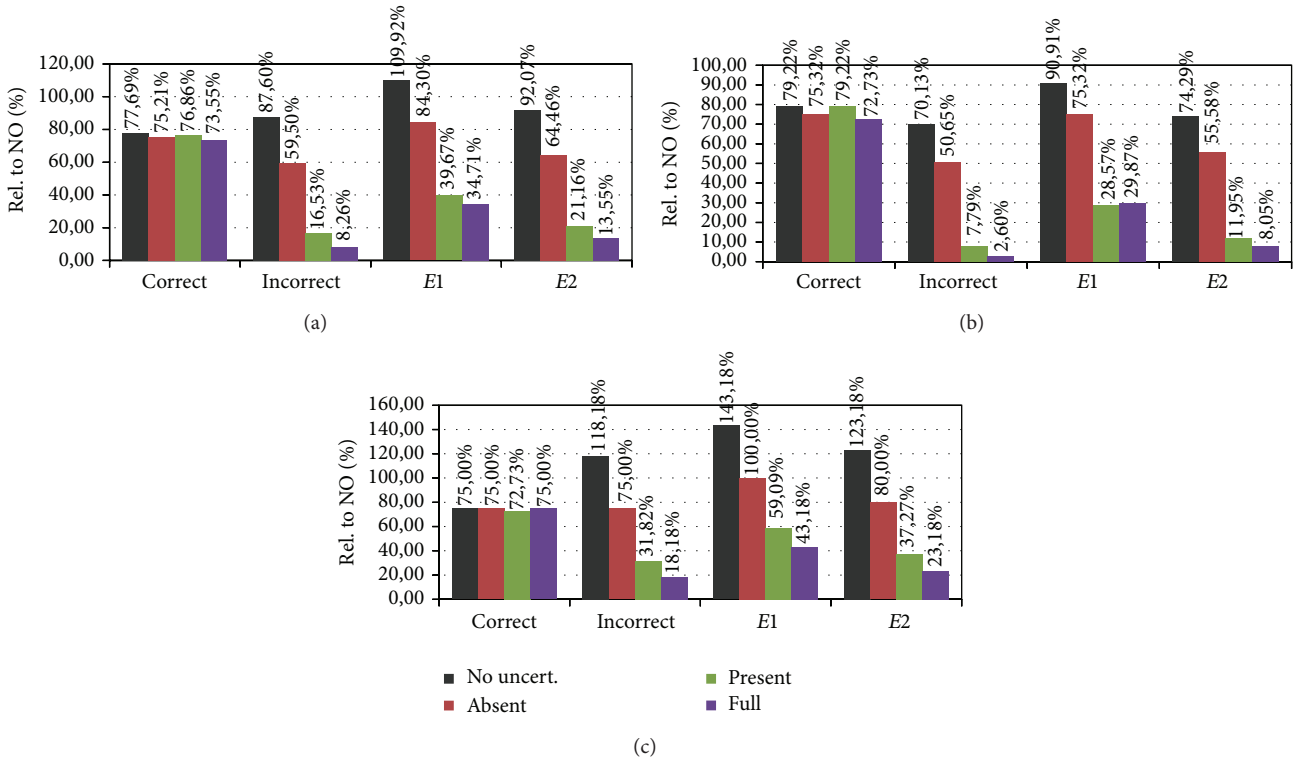


Figure 5: Experiment results statistics for the *combined* (a), *easy* (b) and *hard* (c) test sets. The charts compare the performance parameters (relative to NO—the number of visible known objects) of the system without ignorance awareness (*No uncert.*), with the systems using *absent* part and *present* part ignorance awareness (the first coming from occlusions and field of view and the latter from limited sensor and segmentation precision) as well as the *full* system using both.

recognition system. By applying Dempster-Shafer theory, the uncertainty information has been integrated into a metric-semantic inference system. The sources of uncertainty have been separated into two categories: *absent* object parts, for which the system's belief in their real absence is obtained using a form of occlusion-aware spatial imagination, and *present* object parts, for which the uncertainty is connected to distance-related precision loss and segment size deviations. The impact of both forms of ignorance awareness has been estimated in an experiment involving realistic scenes of an indoor environment with variated difficulty levels. The experiment shows that the use of each kind of uncertainty effectively increases the system reliability by diminishing the false-positive error rates.

The presented inference mechanism is apt to be expanded by adding new kinds of uncertainty, as well as new kinds of evidence concerning the observed scene. In future works we intend to use a wider range of features and focus on real-time performance by exploiting GPU processing power.

Conflict of Interests

The authors declare that there is no conflict of interests regarding the publication of this paper.

Acknowledgment

This work has been partially supported by the National Science Center (Grant 2011/01/B/ST6/07385).

References

[1] P. G. Armour, "The five orders of ignorance," *Communications of the ACM*, vol. 43, no. 10, pp. 17–20, 2000.

[2] A. Varouxaki, N. H. Freeman, D. Peters, and C. Lewis, "Inference neglect and ignorance denial," *British Journal of Developmental Psychology*, vol. 17, no. 4, pp. 483–499, 1999.

[3] J. T. Enns, *The Thinking Eye, The Seeing Brain: Explorations in Visual Cognition*, W. W. Norton & Company, New York, NY, USA, 2004.

[4] G. W. Humphreys, C. J. Price, and M. J. Riddoch, "From objects to names: a cognitive neuroscience approach," *Psychological Research*, vol. 62, no. 2-3, pp. 118–130, 1999.

[5] G. Humphreys and M. Riddoch, "Chapter 15: the cognitive neuropsychology of object recognition and action," in *Handbook of Cognition*, K. Lamberts and R. L. Goldstone, Eds., Psychology Press, Hove, UK, 2001.

[6] B. Harasymowicz-Boggio and B. Siemitkowska, "Object classification with metric and semantic inference," in *Proceedings of the IEEE 6th European Conference on Mobile Robots (ECMR '13)*, pp. 186–191, 2013.

[7] A. P. Dempster and W. F. Chiu, "Dempster-Shafer models for object recognition and classification," *International Journal of Intelligent Systems*, vol. 21, no. 3, pp. 283–297, 2006.

[8] B. Harasymowicz-Boggio and B. Siemi-atkowska, "Object classification using Dempster-Shafer theory," in *Mechatronics 2013: Recent Technological and Scientific Advances*, pp. 559–565, Springer, 2013.

[9] D. F. Wolf, G. S. Sukhatme, D. Fox, and W. Burgard, "Autonomous terrain mapping and classification using hidden Markov models," in *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA '05)*, pp. 2026–2031, April 2005.

[10] S. Thrun, W. Burgard, and D. Fox, *Probabilistic Robotics*, The MIT Press, Cambridge, Mass, USA, 2005.

[11] R. Triebel, B. Frank, J. Meyer, and W. Burgard, "First steps towards a robotic system for exible volumetric mapping of indoor environments," in *Proceedings of the 5th IFAC Symposium on Intelligent Autonomous Vehicles (IAV '04)*, Lisbon, Portugal, 2004.

[12] S. Thrun, W. Burgard, and D. Fox, *Probabilistic Robotics (Intelligent Robotics and Autonomous Agents)*, The MIT Press, Cambridge, Mass, USA, 2005.

[13] E. Remolina and B. Kuipers, "Towards a general theory of topological maps," *Artificial Intelligence*, vol. 152, no. 1, pp. 47–104, 2004.

[14] B. Siemiatkowska, J. Szklarski, M. Gnatowski, and A. Borkowski, "Towards se-mantic navigation system," in *Recent Advances in Intelligent Information Systems*, M. Klopotek, A. Przepiorkowski, S. Wierzchon, and K. Trojanowski, Eds., pp. 711–720, 2009.

[15] R. B. Rusu, Z. C. Marton, N. Blodow, M. Dolha, and M. Beetz, "Towards 3D Point cloud based object maps for household environments," *Robotics and Autonomous Systems*, vol. 56, no. 11, pp. 927–941, 2008.

[16] Ó. Martínez Mozos, R. Triebel, P. Jensfelt, A. Rottmann, and W. Burgard, "Supervised semantic labeling of places using information extracted from sensor data," *Robotics and Autonomous Systems*, vol. 55, no. 5, pp. 391–402, 2007.

[17] R. Lienhart and J. Maydt, "An extended set of Haar-like features for rapid object detection," in *Proceedings of the International Conference on Image Processing (ICIP '02)*, pp. I/900–I/903, September 2002.

[18] H. Zender, O. Mozos, P. Jensfelt, M. Geert-Jan, and W. Burgard, "Conceptual spatial representations for indoor mobile robots," *Robotics and Autonomous Systems (RAS)*, vol. 56, no. 6, pp. 493–502, 2008.

[19] A. Geiger, P. Lenz, and R. Urtasun, "Are we ready for autonomous driving? the KITTI vision benchmark suite," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR '12)*, pp. 3354–3361, Providence, RI, USA, June 2012.

[20] A. Teichman and S. Thrun, "Practical object recognition in autonomous driving and beyond," in *Proceedings of the IEEE Workshop on Advanced Robotics and Its Social Impacts (ARSO '11)*, pp. 35–38, Half-Moon Bay, Calif, USA, October 2011.

[21] T. Goedem, T. Tuytelaars, and L. V. Gool, "Omnidirectional sparse visual path following with occlusion-robust feature tracking," in *Proceedings of the 6th Workshop on Omnidirectional Vision, Camera Networks and Non-classical Cameras (OMNIVIS '05), Conjunction with (ICCV '05)*, 2005, pp. 1806–1811.

[22] A. Ess, K. Schindler, B. Leibe, and L. Van Gool, "Improved multi-person tracking with active occlusion handling," in *Proceedings of the ICRA Workshop on People Detection and Tracking*, vol. 2, 2009.

[23] C. Papazov and D. Burschka, "An efficient RANSAC for 3D object recognition in noisy and occluded scenes," in *Proceedings of the 10th Asian Conference on Computer Vision*, pp. 135–148, Springer, 2010.

[24] M. Z. Zia, M. Stark, and K. Schindler, "Explicit o cclusion modeling for 3D object class representations," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR '13)*, pp. 3326–3333, June 2013.

[25] B. Pepik, M. Stark, P. Gehler, and B. Schiele, "Occlusion patterns for object class detection," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR '13)*, June 2013.

[26] P. L. Bogler, "Shafer-Dempster reasoning with a pplications to multisensor target identification systems," *IEEE Transactions on Systems, Man and Cybernetics*, vol. 17, no. 6, pp. 968–977, 1987.

[27] B. Chokr and V. Kreinovich, "How far are we from complete knowledge, complexity of knowledge acquisition in the Dempster-Shafer approach," in *Advances in the Dempster-Shafer Theory of Evidence*, pp. 555–576, John Wiley & Sons, 1994.

[28] G. Shafer, *A Mathematical Theory of Evidence*, Princeton University Press, Princeton, NJ, USA, 1976.

[29] F. Tombari, S. Salti, and L. Di Stefano, "Unique signatures of histograms for local surface description," in *Proceedings of the 11th European Conference on Computer Vision: Part III (ECCV '10)*, pp. 356–369, Berlin, Germany, 2010.

[30] T. Federico, S. Samuele, and L. D. Stefano, "Unique shape context for 3d data description," in *Proceedings of the ACM Workshop on 3D Object Retrieval (3DOR '10)*, pp. 57–62, New York, NY, USA, 2010.

[31] R. B. Rusu, N. Blodow, and M. Beetz, "Fast point feature histograms (FPFH) for 3D registration," in *Proceedings of the*

IEEE International Conference on Robotics and Automation (ICRA '09), pp. 3212–3217, 2009.

[32] J. Behley, V. Steinhage, and A. B. Cremers, "Performance of histogram descriptors for the classification of 3d laser range data in urban environments," in *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA '12)*, pp. 4391–4398, 2012.

[33] D. Comaniciu and P. Meer, "Mean shift: a robust approach toward feature space analysis," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 24, no. 5, pp. 603–619, 2002.

[34] S. Izadi, D. Kim, O. Hilliges et al., "KinectFusion: Real-time 3D reconstruction and interaction using a moving depth camera," in *Proceedings of the 24th Annual ACM Symposium on User Interface Software and Technology (UIST '11)*, pp. 559–568, New York, NY, USA, October 2011.

Hindawi Publishing Corporation
Mathematical Problems in Engineering
Volume 2014, Article ID 902587, 13 pages
http://dx.doi.org/10.1155/2014/902587

Research Article

Mobile Robots Path Planning Using the Overall Conflict Resolution and Time Baseline Coordination

Yong Ma,[1,2] Hongwei Wang,[2] Langxiong Gan,[1] Min Guo,[2] Liwen Huang,[1] and Jing Zhang[3]

[1] *School of Navigation, Wuhan University of Technology, Hubei Key Laboratory of Inland Shipping Technology, Wuhan, Hubei 430063, China*
[2] *Key Laboratory of Image Processing and Intelligent Control, Institute of Systems Engineering, Huazhong University of Science and Technology, Wuhan, Hubei 430074, China*
[3] *School of Business, Chutian College, Huazhong Agricultural Business University, Wuhan, Hubei 430205, China*

Correspondence should be addressed to Langxiong Gan; glx701227@163.com

Received 5 January 2014; Revised 17 April 2014; Accepted 1 May 2014; Published 22 June 2014

Academic Editor: Huosheng Hu

This paper aims at resolving the path planning problem in a time-varying environment based on the idea of overall conflict resolution and the algorithm of time baseline coordination. The basic task of the introduced path planning algorithms is to fulfill the automatic generation of the shortest paths from the defined start poses to their end poses with consideration of generous constraints for multiple mobile robots. Building on this, by using the overall conflict resolution, within the polynomial based paths, we take into account all the constraints including smoothness, motion boundary, kinematics constraints, obstacle avoidance, and safety constraints among robots together. And time baseline coordination algorithm is proposed to process the above formulated problem. The foremost strong point is that much time can be saved with our approach. Numerical simulations verify the effectiveness of our approach.

1. Introduction

Since the 1990s, industrial robots have played an important role in the robotics research [1]. One foremost research topic of industrial robots is mobile robots' path planning [2, 3], wherein the paths connecting the start and end poses of robots are planned to assist completing their task. In industrial application, to suit the actual situation, there is no denial that the nonholonomic constraint [4, 5] which means the mobile robots subjecting to the requirements of rolling without slipping should be taken into consideration. Further, in the traverse process of mobile robots, owing to the mutual influence of robots and obstacles moving and stationary in a time-varying environment, the conflicts would be induced inevitably. Thus, our concerned path planning problems become much more difficult since the planned paths must take above geometric and nonholonomic constraints and conflict-avoidance constraints into account. Then, how to plan the shortest paths for multiple mobile robots while explicitly considering smoothness, kinematics constraints, motion boundary, obstacle avoidance, and safety constraints among mobile robots becomes our mainly focused problem.

During the past decades, various path planning methodologies for mobile robots have been developed [2, 6–12]. We have presented the detailed introduction in [2] with a sense of hierarchy. In [6, 13], differential evolution and honey bee mating optimization algorithms are used to resolve the cooperative multirobot path planning problem, respectively. However, too many broken lines exist in the planned paths, which cannot be directly used in the real scenario. With those ill-smooth paths as depictured in [14], the roller slip would be triggered in the motion process of mobile robots. Then, the slip would endanger the safety of mobile robots and furthermore lead to the meaninglessness of those paths.

As pointed in [15] the planned path should be smooth to accord with the practical applications. With the smooth paths, we can account for the continuity [16] and the kinematics constraints sufficiently. Contrarily, if with the ill-smooth paths, the jerks would be triggered during the motion process. Given that polynomials are the set of highly smooth

functions, there is no doubt that polynomial based paths are quite smooth [15]. Hence, we adopt the polynomial based paths to represent their motion in our research.

Generally in many works like [2, 9, 10], they focus on two-phase approach for path planning problems. Firstly, the path for each mobile robot is reached with some certain algorithm. Following that, by using the predefined principles, the conflicts among the planned paths are resolved in the second phase. However, approaches of this category cannot be illustrated analytically and do not work sometimes. Consequently, the sound approach is disposing of all the conflicts as a whole not using the two-phase approach.

To avoid the two-phase methods, the studies like [2, 7–11] adopted quartic Bezier curves to represent the paths of mobile robots. However, approaches in [7, 8] strictly restrict the paths to the fourth-order Bezier curves and require the initial and final velocities of each mobile robot, which would be impractical for the realistic application. Comparatively, as a natural continuation of our previous work in [2], to avoid the deadlock of the two-phase conflict dissolution approach, in our objective function, we eliminate the overall conflicts of the focused problem. When it comes to the objective function, we set the summation of the shortest paths of all mobile robots as the goal function and take generous constraints into account. Eventually, we formulate the problem as one time-varying nonlinear programming problem (TNLPP). To solve the TNLPP, we devise the time baseline coordination method. Within this method, we choose the least motion time of all mobile robots as the time baseline; by using this parameter we can fulfill their motion coordination. The detailed illustration is presented in Section 4.

This paper is organized into seven sections. In Section 2, the shortest polynomial based paths for multiple mobile robots are defined. In Section 3, modeling for path planning problems with overall confliction is introduced explicitly. In Section 4, time baseline coordination algorithm is detailed to resolve the path planning problems. In Section 5, simulation results are presented to demonstrate the effectiveness of our approaches. Section 6 discusses results of cases in Section 5. Section 7 concludes this paper.

2. Problem Formulation

Suppose that one mobile robot A_i $(i = 1, 2, \ldots, J)$ is represented by a r_i-radius circle with center $x_i(t) = (x_{1i}(t), x_{2i}(t))$ and orientation $\theta_i(t)$, $t \in [0, t_{fi}]$, where m is the number of mobile robots, x_i is one parameterized polynomial with unknown index and used to denote the path of A_i, and t_{fi} is a given real number as final moving time of A_i. The poses of mobile robots at time t are illustrated as $P_i(t) = (x_i(t), \theta_i(t))$. Then, we should plan the shortest and conflict-free paths for all mobile robots A_i $(i = 1, 2, \ldots, J)$ from the start pose P_{i0} to the end pose P_{if}, where $P_{i0} = P_i(0)$ and $P_{if} = P_i(t_f)$.

As sensors or cameras are used to detect the positions of obstacles [17], the paths of obstacles can be duly reached. Meanwhile, suppose the obstacle O_k $(k = 1, 2, \ldots, q)$ is a r_k-radius circle with center $\alpha_k(t) = (\alpha_{1k}(t), \alpha_{2k}(t))$, $t \in [0, t_k]$, where q is the number of the obstacles, $\alpha_k(\cdot)$ is the path of O_k and expressed with known parameterized polynomial, and t_k is a given real number as final motion time of O_k.

2.1. Polynomial Based Paths. To detail x_i which is the path of A_i and equals $(x_{1i}(t), x_{2i}(t))$, we set $x_{1i}(t) = x_{1i}$, $x_{2i}(t) = x_{2i}$, and the unknown indices of x_{1i} and x_{2i} are M and N, respectively. The unknown coefficients of x_{1i} and x_{2i} are represented by $[a_{0i}\ a_{1i}\ \cdots\ a_{mi}\ \cdots\ a_{Mi}]$ and $[b_{0i}\ b_{1i}\ \cdots\ b_{ni}\ \cdots\ b_{Ni}]$ $(1 \le m \le M, 1 \le n \le N)$, respectively. Then, the path of A_i is given in the following form, where $t \in [0, t_{fi}]$:

$$\begin{aligned} x_{1i} &= a_{0i} + a_{1i}t + \cdots + a_{mi}t^m + \cdots + a_{Mi}t^M, \\ x_{2i} &= b_{0i} + b_{1i}t + \cdots + b_{ni}t^n + \cdots + b_{Ni}t^N. \end{aligned} \tag{1}$$

Suppose that $A^i = [a_{0i}\ a_{1i}\ \cdots\ a_{mi}\ \cdots\ a_{Mi}]$, $B^i = [b_{0i}\ b_{1i}\ \cdots\ b_{ni}\ \cdots\ b_{Ni}]$, $T_A^i = [t^0\ t\ \cdots\ t^m\ \cdots\ t^M]$, and $T_B^i = [t^0\ t\ \cdots\ t^n\ \cdots\ t^N]$, $t \in [0, t_{fi}]$. Then, the path x_i can be

$$\begin{bmatrix} x_{1i} \\ x_{2i} \end{bmatrix} = \begin{bmatrix} A^i & 0 \\ 0 & B^i \end{bmatrix} \begin{bmatrix} {T_A^i}^T \\ {T_B^i}^T \end{bmatrix}. \tag{2}$$

Within path x_i, the velocities in the direction of x_{1i} and x_{2i} are $\dot{x}_{1i}(t)$ and $\dot{x}_{2i}(t)$, respectively. The accelerations in the direction of x_{1i} and x_{2i} are $\ddot{x}_{1i}(t)$ and $\ddot{x}_{2i}(t)$, respectively. Consequently, the path length of A_i can be

$$s_i = \int_0^{t_{fi}} \sqrt{\dot{x}_{1i}^2(t) + \dot{x}_{2i}^2(t)}\, dt = \int_0^{t_{fi}} \left\| \dot{x}_i(t) \right\|_2 dt. \tag{3}$$

Analogously, for the path of O_k featured by α_k, we set $\alpha_{1k}(t) = \alpha_{1k}, \alpha_{2k}(t) = \alpha_{2k}$, and the known indices of α_{1k} and α_{2k} are L and U, respectively. The known coefficients of α_{1k} and α_{2k} are represented by $[a_{0Ok}\ a_{1Ok}\ \cdots\ a_{lOk}\ \cdots\ a_{LOk}]$ and $[b_{0Ok}\ b_{1Ok}\ \cdots\ b_{uOk}\ \cdots\ a_{UOk}]$ $(1 \le l \le L, 1 \le u \le U)$, respectively. Then the path of O_k is given in the following form, where $t \in [0, t_{fi}]$:

$$\begin{aligned} \alpha_{1k} &= a_{0Ok} + a_{1Ok}t + \cdots + a_{lOk}t^l + \cdots + a_{LOk}t^L, \\ \alpha_{2k} &= b_{0Ok} + b_{1Ok}t + \cdots + b_{uOk}t^u + \cdots + b_{UOk}t^U. \end{aligned} \tag{4}$$

Suppose that $A_O^k = [a_{0Ok}\ a_{1Ok}\ \cdots\ a_{lOk}\ \cdots\ a_{LOk}]$, $B_O^k = [b_{0Ok}\ b_{1Ok}\ \cdots\ b_{uOk}\ \cdots\ a_{UOk}]$, $T_A^k = [t^0\ t\ \cdots\ t^l\ \cdots\ t^L]$, and $T_B^k = [t^0\ t\ \cdots\ t^u\ \cdots\ t^U]$, $t \in [0, t_k]$. Then, the path α_k can be

$$\begin{bmatrix} \alpha_{1k} \\ \alpha_{2k} \end{bmatrix} = \begin{bmatrix} A_O^k & 0 \\ 0 & B_O^k \end{bmatrix} \begin{bmatrix} {T_A^k}^T \\ {T_B^k}^T \end{bmatrix}. \tag{5}$$

2.2. Generous Constraints. To plan the effective and reasonable paths for multiple mobile robots, when achieving our objective function, generous constraints including kinematics, motion boundaries, and obstacle avoidance should be satisfied. For this end, to probe into the essence of each constraint, we detail each constraint sequentially.

2.2.1. Kinematics Constraints. Kinematics constraints indicate that the velocity and acceleration along with the path should be bound to their maximum allowed intervals to prevent robots from slipping. Then, the kinematics constraints can be

$$v_i(t) \le v_{\max i}, \quad a_i(t) \le a_{\max i}, \quad t \in \left[0, t_{fi}\right], \ i \in [1, J]. \tag{6}$$

In the kinematics constraints, $v_i(t)$ and $a_i(t)$ are denoted by $\dot{x}_i(t)$ and $\ddot{x}_i(t)$, respectively.

2.2.2. Motion Boundary. X is used to represent the two-dimensional environment boundary. The constraints imposed by the motion boundary should be satisfied as below:

$$\begin{aligned} &P_i(0) = (x_{0i}, \theta_{0i}), \\ &P_i\left(t_{fi}\right) = \left(x_{fi}, \theta_{fi}\right), \\ &X_{0i}, X_{fi} \in X, \\ &\qquad i \in [1, J]. \end{aligned} \tag{7}$$

2.2.3. Obstacle Avoidance Constraints. Obstacle avoidance constraints guarantee the least safe distance between the current moving robot and all the obstacles. Suppose that d_{oik} and $d_{\text{obs\_}ik}$ are the real distance and the preset safe distance between A_i and O_k, respectively. d_{oik} is represented as below. For $1 \le i \le J$, $1 \le k \le q$,

$$d_{oik}(t) = \left\| x_i(t) - \alpha_k(t) \right\|_2 - \left(r_i + r_k\right). \tag{8}$$

Then, the obstacle avoidance constraints for A_i and O_k $(1 \le i \le J, 1 \le k \le q)$ in X should be

$$d_{\text{obs\_}ik} - d_{oik}(t) \le 0, \quad t \in \left[0, \max\left(t_{fi}, t_k\right)\right]. \tag{9}$$

2.2.4. Safety Constraints among Mobile Robots. Within the safety constraints among mobile robots, all the robots can be kept away from each other with distances larger than the predefined safety value. Thus, the collisions between mobile robots can be averted triumphantly with ease.

Suppose d_{ij} and d_{s_ij} are the real distance and the preset safe distance between A_i and A_j, respectively. For $1 \le i \le J$, $i \ne j$, d_{ij} can be

$$d_{ij}(t) = \left\| x_i(t) - x_j(t) \right\|_2 - \left(r_i + r_j\right). \tag{10}$$

Then, the safety constraints among A_i in X should be satisfied as below $(1 \le i \le J)$:

$$d_{s\_ij} - d_{ij}(t) \le 0, \quad t \in \left[0, \max\left(t_{fi}, t_{fj}\right)\right]. \tag{11}$$

2.3. Goal Function. Our goal function is to minimize the summation of all the paths length, which can be

$$\min \sum_{i=1}^{J} s_i. \tag{12}$$

Similar to our work in [2], we equally divide the time interval $[0, t_{fi}]$ into F parts, where h denotes the value of each part, $h = t_{fi}/F$, $0 \le f \le F$; set $E_i(t) = \|\dot{x}_i(t)\|_2$. Consequently, after the transformation, the goal function can be

$$\min \sum_{i=1}^{J} \frac{h\left(2\sum_{f=1}^{F} E_i(fh) - E(0) - E_i\left(t_{fi}\right)\right)}{2}. \tag{13}$$

3. Problem Modeling with the Overall Conflict Resolution

3.1. The Meaning of the Overall Conflict Resolution. Generally in the path planning problems, the conflicts including the collisions among mobile robots owing to the crossover of their planned paths and the crashes between mobile robots and obstacles occurred within the motion boundary. To fulfill the conflict-free path planning, the strategies or methods adopted to remove and avoid the collisions can be named conflict resolution strategies. Therein, we propose the overall conflict resolution.

In our overall conflict resolution, we consider all the constraints demonstrated above in the process of modeling. By using the overall conflict resolution, the phase of how to bring forward some ideal conflict resolution strategies in the multiphase approaches is deleted. However, compared with the path planning model of the multiphase methods, the founded model based on overall conflict resolution is much more difficult and harder to actualize.

3.2. Problem Modeling. As none of the above constraints can be violated, with overall conflict resolution, for $1 \le i, j \le J$, $i \ne j$, $1 \le k \le q$, we consider all the constraints as a whole in our objective function as below:

$$\begin{aligned} \min \quad & \sum_{i=1}^{J} \frac{h\left(2\sum_{f=1}^{F} E_i(fh) - E(0) - E_i\left(t_{fi}\right)\right)}{2} \\ \text{s.t.} \quad & v_i(t) \le v_{\max i} \quad t \in \left[0, t_{fi}\right] \\ & a_i(t) \le a_{\max i} \quad i \in [1, J] \\ & P_i(0) = (x_{0i}, \theta_{0i}) \\ & P_i\left(t_{fi}\right) = \left(x_{fi}, \theta_{fi}\right) \\ & X_{0i}, X_{fi} \in X \\ & \qquad i \in [1, J] \\ & d_{s\_ij} - d_{ij}(t) \le 0, \quad t \in \left[0, \max\left(t_{fi}, t_{fj}\right)\right] \\ & d_{\text{obs\_}ik} - d_{oik}(t) \le 0, \quad t \in \left[0, \max\left(t_{fi}, t_k\right)\right]. \end{aligned} \tag{14}$$

Based on the overall conflict resolution strategy, our focused path planning problem is transformed to one objective function while subjects to all above constraints. Substantially, the concerned problem is one time-varying nonlinear programming problem with multiple nonlinear constraints. In (14), conflicts among all the mobile robots and obstacles

Input: Start and end poses P_{i0} and P_{if}, respectively. Unknown path $x_i(t)$, a known path $\alpha_k(t)$, motion time t_{fi} and t_k, and number of divided time interval F, temporary time variable $T_{\min}$.
Output: Parameters A^i, B^i, T_A^i and T_B^i which are used to represent the path of A_i.
Begin
$T_{\min} = 0$;
foreach t_{fi} of A_i **do**
$T_{\min} = \min t_{\text{fi}}$;
end foreach
foreach A_i in the set of multiple mobile robots
Map motion time interval $[0, t_{\text{fi}}]$ of A_i into $[0, t_{\text{fi}}/T_{\min}]$ with the multiples of $1/T_{\min}$.
Map constraints of kinematics and obstacle-avoidance of A_i into corresponding forms with respect to $[0, t_{\text{fi}}/T_{\min}]$.
end foreach
Modify the objective function with all the adjustments, then reach the newly TNLPPs named $\text{TNLPPs}_{\text{\_new}}$.
Call $\text{Trap}_{\text{\_TNLPPs}}(\text{TNLPPs}_{\text{\_new}})$
foreach A_i **do**
Multiples $A^{i\prime}$, $B^{i\prime}$, $T_A^{i\prime}$ and $T_B^{i\prime}$ with $T_{\min}$.
end foreach
return A^i, B^i, T_A^i and T_B^i.
End

Algorithm 1: $\text{Time}_{\text{Coor}}(P_{i0}, P_{if}, x_i(t), t_{fi}, \alpha_k(t), t_k, F, T_{\min})$.

moving and static are considered explicitly, which evades the general multilevel conflicts. Thus, compared with the multiphrase approaches, since path planning for all mobile robots just needs to be calculated only once, it is evident that much time can be saved by using our approach. Logically, with this model, the path planning efficiency is improved evidently. However, all above advantages are gained at the cost of the upgrading of the resolve difficulty of our model. To entirely dissolute those conflicts, in the following section we propose the time baseline coordination approach.

4. Optimal Solution to Path Planning Using the Time Baseline Coordination

In the path planning problems, suppose that all the mobile robots commence moving at the same time. In this section, we illustrate the time baseline coordination at first, following that analyze the equivalence of this approach, and finally put forward the implementation of our subtle approach.

4.1. Brief Introduction of the Time Baseline Coordination. As depicted in Section 2, the final motion time of A_i and O_k is t_{fi} and t_k, respectively. Suppose that the baseline time is $T_{\min}$. The essence of our time baseline coordination is selecting the least motion time of A_i denoted as $T_{\min}$ to coordinate the motion of all the mobile robots.

In the time baseline coordination, we demonstrate the general operating procedures concisely.

Firstly, with $T_{\min}$, we can transform all the velocities, the accelerations, and so forth into $[0, t_{fi}/T_{\min}]$.

Following that, we renew the corresponding constraints in our goal function.

Thirdly, using the algorithm of our former work in [2], we solve then the transformed problems.

And then, with multiple $1/T_{\min}$, we map the resolution into $[0, t_{fi}]$ corresponding to the first step.

Finally, we can reach the final solution eventually.

4.2. Time Baseline Coordination Algorithm Time_{Coor}. To illustrate the proposed approach definitely, we devise the following time baseline coordination algorithm called $\text{Time}_{\text{Coor}}$ (see Algorithm 1).

The algorithm $\text{Trap}_{\text{\_TNLPPs}}$ is our previous work in [2], which is proposed to solve the time-varying nonlinear programming problem. Brief flow of algorithm $\text{Trap}_{\text{\_TNLPPs}}$ is presented in the Appendix, and a comprehensive procession can be found in [2].

4.3. Equivalence of the Time Baseline Coordination Algorithm. To proof the equivalence of our approach, several equivalent descriptions (which are abbreviated as Equ) are presented as below.

Equ-1. The motion boundary has not varied after mapping.

Proof. X is the given motion boundary, which has little relation with the time. □

Equ-2. The ration of the transformed velocities to the former is $T_{\min}$.

Proof. As shown in Equ-1, there is no change in the trajectory of A_i, while the motion time is mapped into $[0, t_{fi}/T_{\min}]$. Thus, with theorem of integral, the division between the current velocities and the former is $T_{\min}$. □

Equ-3. The ration of the transformed accelerations to the former is $T_{\min}^2$.

Proof. Based on Equ-1 and Equ-2, according to the relationship between velocities and accelerations, with theorem of integral, the division between the current accelerations and the former is $(T_{\min})^2$. □

Equ-4. The ration of the velocities in the final solution to the velocities in the mapped solution is $1/T_{\min}$.

Proof. Based on Equ-1 and Equ-2, there is no change in the trajectory of A_i, while the motion time is transformed from $[0, t_{fi}/T_{\min}]$ to $[0, t_{fi}]$. With theorem of integral, the division between the velocities in the final solution and the velocities in the mapped solution is $1/T_{\min}$. □

Equ-5. The ration of the accelerations in the final solution to the accelerations in the mapped solution is $1/(T_{\min})^2$.

Proof. Based on Equ-1 and Equ-2, according to the relationship between velocities and accelerations, with theorem of integral, the division between the accelerations in the final solution and the accelerations in the mapped solution is $(T_{\min})^2$. □

With the overall conflict resolution, when all constraints are considered, paths for all mobile robots are promptly achieved based on algorithm $\text{Time}_{\text{Coor}}$.

5. Simulation

Simulations are carried out to show the effectiveness of our approach. Four curves have been used increasingly as metrics to evaluate the performance of the proposed approach [2, 7, 8, 18]. They are the curve of velocities relative to time, the curve of acceleration relative to time, the curve of the safety distances relative to time, and the curve of planned paths relative to their motion time. Experiments are conducted under several different time-varying scenarios.

In the simulation, we first consider the case without obstacles and then the latter with static and moving obstacles. In the following figures of Section 5, the scales are same and all quantities conform to one certain unit system, for instance, meters and meters per second.

5.1. Path Planning for Multiple Mobile Robots without Obstacles. When it comes to the case of multiple mobile robots moving synchronously in the environment without obstacles, for $1 \le i, j \le J, i \ne j$, the essential objective function can be

$$\begin{aligned}
\min \quad & \sum_{i=1}^{J} \frac{h\left(2\sum_{f=1}^{F} E_i(fh) - E(0) - E_i\left(t_{fi}\right)\right)}{2} \\
\text{s.t.} \quad & v_i(t) \le v_{\max i} \quad t \in \left[0, t_{fi}\right] \\
& a_i(t) \le a_{\max i} \quad t \in \left[0, t_{fi}\right] \\
& P_i(0) = \left(x_{0i}, \theta_{0i}\right) \\
& P_i\left(t_{fi}\right) = \left(x_{fi}, \theta_{fi}\right) \\
& X_{0i}, X_{fi} \in X \\
& \qquad i \in [1, J] \\
& d_{s_ij} - d_{ij}(t) \le 0, \quad t \in \left[0, \max\left(t_{fi}, t_{fj}\right)\right].
\end{aligned} \tag{15}$$

Table 1: Initial information of mobile robots A_i $(i = 1, 2, 3)$ in Section 5.1.1.

A_i	P_{i0}	P_{if}	v_0	v_f	$v_{\max}$
A_1	$[0.2, 1.4, -\pi/4]^T$	$[1.4, 0.2, -\pi/4]^T$	0.4	0.4	0.8
A_2	$[1.4, 0.2, 3\pi/4]^T$	$[0.2, 1.4, 3\pi/4]^T$	0.4	0.5	0.8
A_3	$[0.2, 0.2, \pi/4]^T$	$[1.4, 1.4, \pi/4]^T$	0.4	0.4	0.8

5.1.1. Path Planning for Three Mobile Robots without Obstacles. In the first case, to test the validity of our approach, we choose the data listed in Table 1 from recently published work [8] to examine the effectiveness of our algorithm. Approach in [8] was based on fourth-order Bezier curve.

In [8], the safety distances, the maximum motion time, and the maximum accelerations are given as $d_{s\_ij} = 0.35$, $T_{\max i} = 5$, and $a_{\max i} = 0.5$ $(i, j = 1, 2, 3, i \ne j)$, respectively.

Within our approach, in order to fulfill the coordination, the minimized motion time of all the mobile robots A_i $(i = 1, 2, 3)$ is chosen as the time baseline which is used for coordination. Subsequently, applying the coordination algorithm $\text{Time}_{\text{Coor}}$, for $t \in [0, 5]$, paths of A_i $(i = 1, 2, 3)$ are obtained as below:

$$\begin{aligned}
A^1 &= [0.2 \;\; 0.2828 \;\; -0.166 \;\; 0.097 \;\; -0.021 \;\; 0.001]; \\
B^1 &= [1.4 \;\; -0.2828 \;\; 0.198 \;\; -0.073 \;\; 0.007]; \\
A^2 &= [1.4 \;\; -0.2828 \;\; 0.518 \;\; -0.052 \;\; 0.017 \;\; -0.002]; \\
B^2 &= [0.2 \;\; 0.283 \;\; -0.127 \;\; 0.041 \;\; -0.004]; \\
A^3 &= [0.2 \;\; 0.2828 \;\; 0.216 \;\; -0.116 \;\; 0.019 \;\; -0.001]; \\
B^3 &= [1.4 \;\; 0.2828 \;\; 0.145 \;\; -0.065 \;\; 0.007]; \\
T_A^i &= [1 \;\; t_i \;\; t_i^2 \;\; t_i^3 \;\; t_i^4 \;\; t_i^5]; \\
T_B^i &= [1 \;\; t_i \;\; t_i^2 \;\; t_i^3 \;\; t_i^4 \;\; t_i^5].
\end{aligned}$$

Corresponding to Figures 1(a), 1(b), and 1(c), our planned paths evolve from the perspective of three different time intervals [0, 2.3], [2.3, 5], and [0, 5]. In Figures 1(a), 1(b), and 1(c), it is shown that each mobile robot smoothly moves from its initial pose to final pose without any collision. The paths of [8] in the time interval [0, 5] are shown in Figure 1(d). Thus, our approach based on $\text{Time}_{\text{Coor}}$ can manipulate the multiple mobile robots path planning problems successfully.

The variation curves of velocities and accelerations along the time are shown in Figures 2(a) and 2(b), respectively. Curves in Figure 2 are sufficient to the constraints in Table 1 and all other requirements in [8] fully. Then the planned paths meet the constraints of kinematics.

Figure 3 illustrates that the variation of safety distances d_{12}, d_{13}, and d_{23} is satisfied with the predefined requirements in this case.

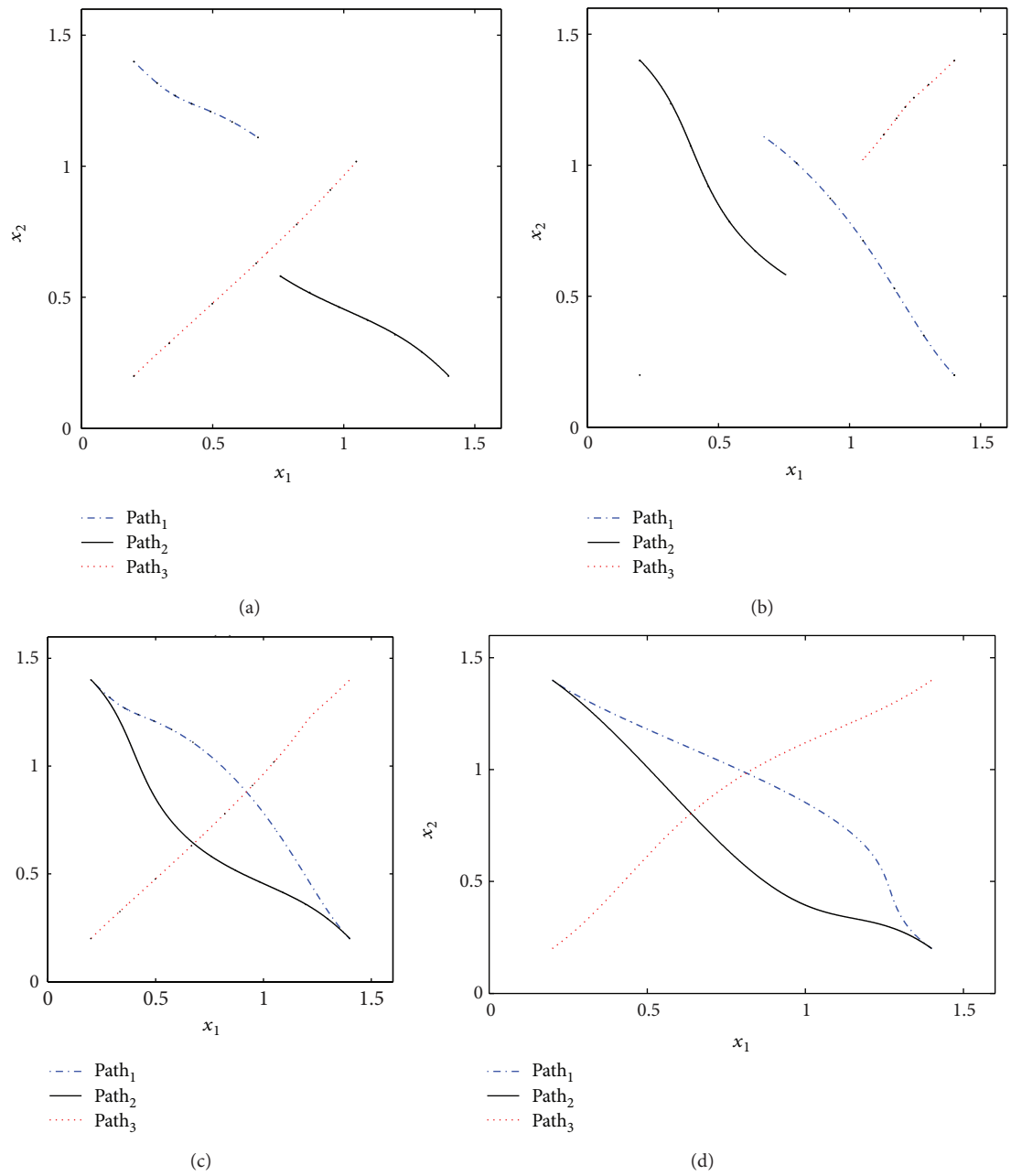
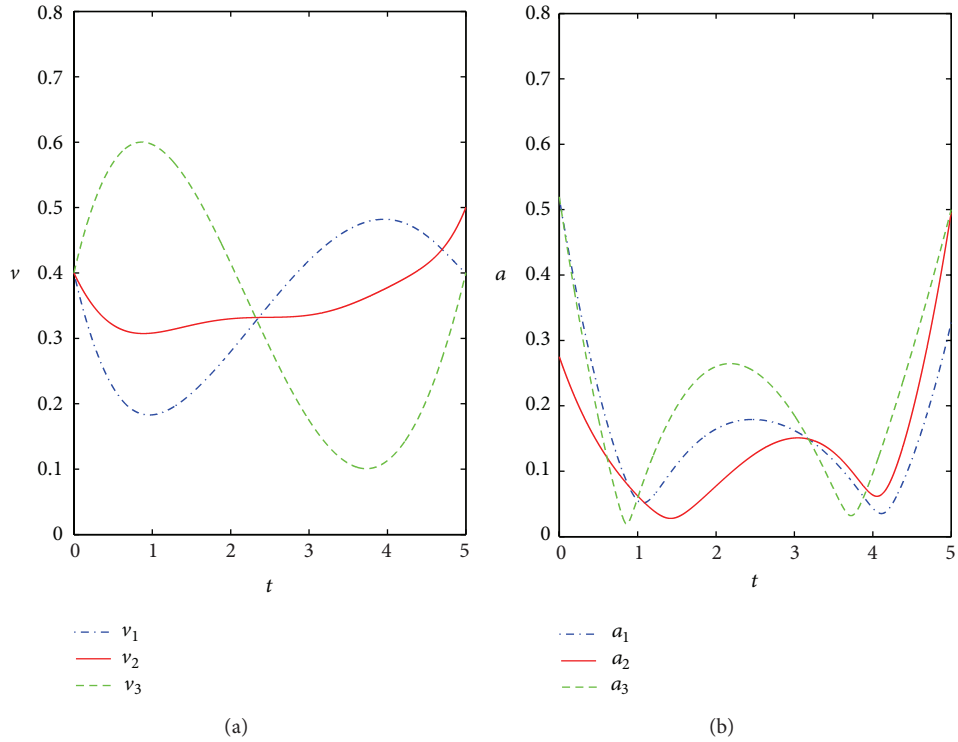


Figure 1: Planned paths evolvement of A_i and paths of [8], $i = 1, 2, 3$. (a) In [0, 2.3], (b) in [2.3, 5], (c) in [0, 5], and (d) in [0, 5].

In our approach, the polynomials are general that are more flexible than the polynomials of [8] which are limited to four. Meanwhile, with approach in [8], prior to path planning, the initial and final velocities of each mobile robot should be specified in detail. Generally, in practice, the velocities belong to some certain intervals, not the constant predefined value. Contrastively, our approach is much more suitable for the general multiple mobile robots path planning problems.

Compared to our prior work in [2], the paths of A_i ($i = 1, 2, 3$) should be calculated for three times in the path planning phase. In our approach, it only takes a third of the time in the first phase of [2] to reach the conflict-free solution. Furthermore, the planned paths of A_1 and A_2 in [2] are almost overlapped, and the conflicts would exist in the planned paths of A_i ($i = 1, 2, 3$); the phase of conflict coordination should be applied which can be ineluctable. Synthetically, from the perspective of calculational time, the proposed approach excels the approach of [2] remarkably.

5.1.2. Path Planning for Four Mobile Robots without Obstacles. To validate the effectiveness and feasibility of our algorithm under much more complex time-varying environment, the simulation scenario is presented with four mobile robots moving synchronously bound to the limited area. For this case, the essential objective function can be (15), wherein the



(a) (b)

Figure 2: (a) v-t, (b) a-t in Section 5.1.1.

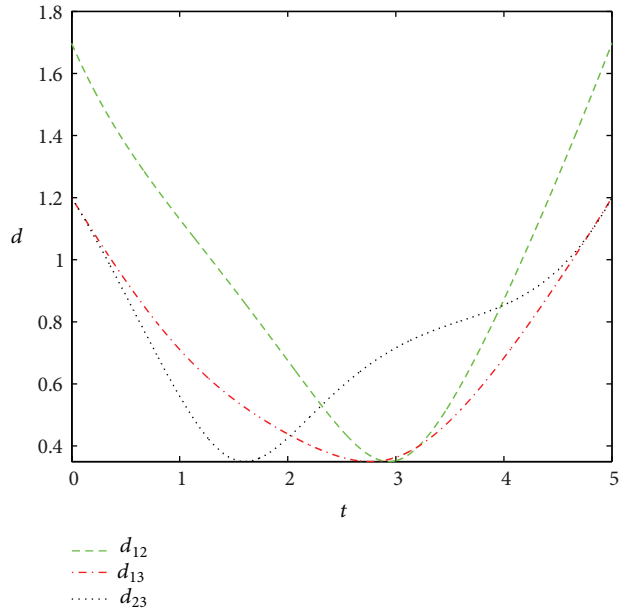


Figure 3: Variation curve of safety distance d-t in Section 5.1.1.

maximum number of the robots equals 4 and the test data is listed in Table 2.

The preset safety distances are $d_{s\_ij} = 0.25$ $(i = 1, j = 2, 3)$, $d_{s\_ij} = 0.25$ $(i = 2, j = 3, 4)$, $d_{s\_14} = 0.035$, and $d_{s\_34} = 0.25$.

In the time baseline coordination method, we select the minimized motion time of A_i $(i = 1, 2, 3, 4)$ as the coordinate factor in the function (15). Then, after applying

Table 2: Initial information of mobile robots A_i $(i = 1, 2, 3, 4)$ in Section 5.1.2.

A_i	P_{i0}	P_{if}	$T_{\max}$	$v_{\max}$	$a_{\max}$
A_1	$[0.8, 0.8, \pi/4]^T$	$[0.2, 0.2, -3\pi/4]^T$	5	0.8	0.5
A_2	$[1.4, 0.2, 3\pi/4]^T$	$[0.2, 1.4, \pi/2]^T$	5	0.8	0.5
A_3	$[0.2, 1.4, -\pi/4]^T$	$[1.4, 0.2, 0]^T$	5	0.8	0.5
A_4	$[0.2, 0.2, \pi/4]^T$	$[1.4, 1.4, \pi/4]^T$	5	0.8	0.5

the coordination algorithm $\text{Time}_{\text{Coor}}$, the paths of A_i $(i = 1, 2, 3, 4)$ are reached as below:

$$x_{11}(t) = 0.8 - 0.2819t + 0.0407t^2 - 0.00167t^3,$$

$$x_{21}(t) = 0.8 - 0.2689t + 0.03t^2 + 0.00141t^3 - 0.000257t^4, \quad t \in [0, 5],$$

$$x_{12}(t) = 1.4 - 0.291t - 0.00735t^2 + 0.0035t^3,$$

$$x_{22}(t) = 0.2 + 0.472t - 0.04731t^2 - 0.00136t^3 + 0.000311t^4, \quad t \in [0, 5],$$

$$x_{13}(t) = 0.2 + 0.149t + 0.0044t^2 + 0.00276t^3,$$

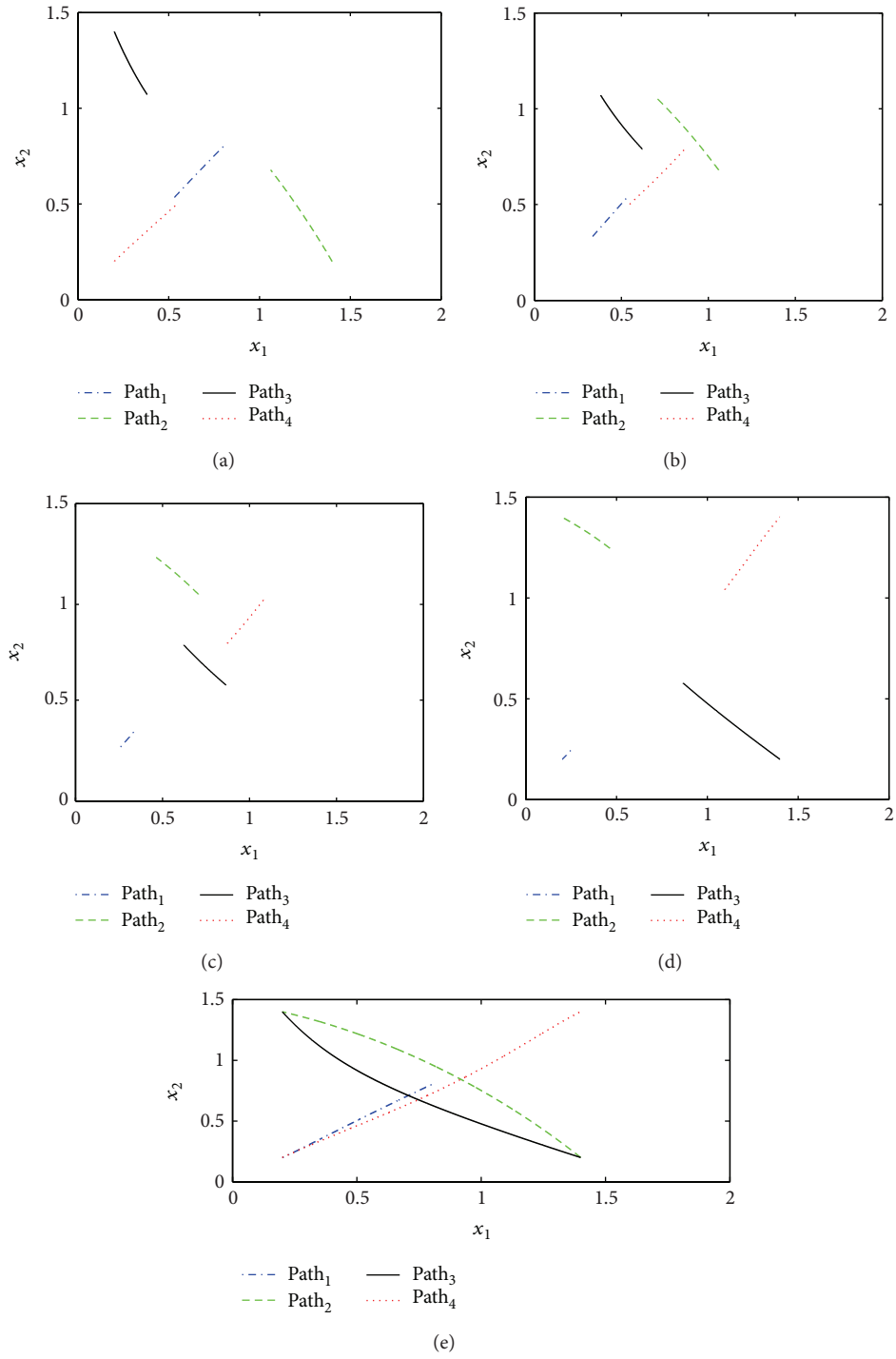


FIGURE 4: Planned paths evolvement of A_i $(i = 1, 2, 3, 4)$ in Section 5.1.2. (a) In $[0, 1.15]$, (b) in $[1.15, 2.4]$, (c) in $[2.4, 3.4]$, (d) in $[3.4, 5]$, and (e) in $[0, 5]$.

$$x_{23}(t) = 1.4 - 0.3353t + 0.0513t^2 - 0.00812t^3 + 0.00036t^4, \quad t \in [0, 5],$$

$$x_{14}(t) = 0.2 + 0.326t - 0.02121t^2 + 0.00079t^3,$$

$$x_{24}(t) = 0.2 + 0.3003t - 0.048t^2 + 0.01446t^3 - 0.001453t^4, \quad t \in [0, 5]. \tag{16}$$

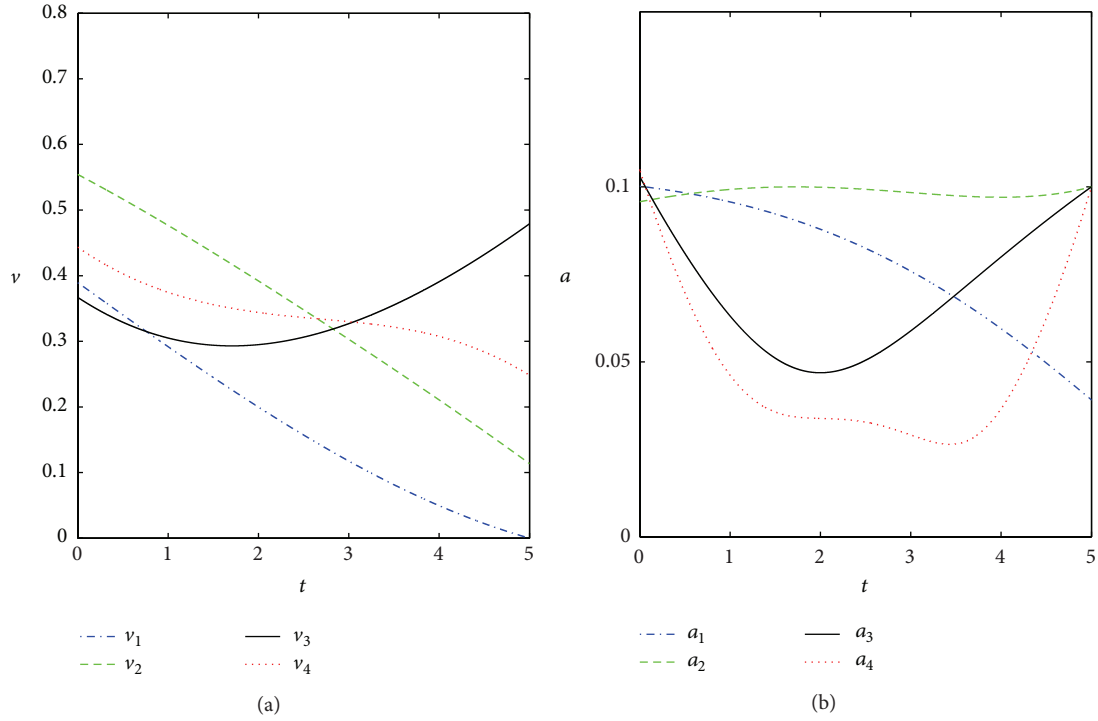


FIGURE 5: (a) v-t, (b) a-t in Section 5.1.2.

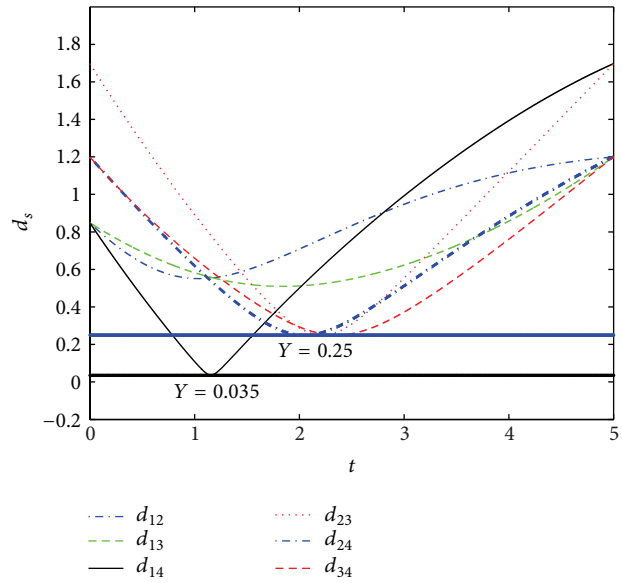


FIGURE 6: Variation curve of safety distance d-t in Section 5.1.2.

Figure 4 demonstrates the paths evolvement from the perspective of five different time intervals [0, 1.15], [1.15, 2.4], [2.4, 3.4], [3.4, 5], and [0, 5]. In Figure 4, the subfigures from (a) to (e) show that each mobile robot smoothly moves from its start pose to end pose without any collision.

The variation curves of velocities and acceleration along with time are shown in Figures 5(a) and 5(b), respectively. Curves in Figure 5 are sufficient to the requirements of the constraints in Table 2 fully. Thus, the planned paths are consistent with the constraints of kinematics.

Furthermore, Figure 6 illustrates that the variation of safety distances $d_{12}, d_{13}, d_{14}, d_{23}, d_{24}$, and d_{34} lives up to the predefined requirements quite well.

In a word, with above verification, it is substantiated that our approach is effective for path planning problems with even much more time-varying environment.

5.2. Path Planning for Mobile Robots with Static and Moving Obstacles. The path planning optimization function for multiple mobile robots with static and moving obstacles is the same as (14). To test the effectiveness of algorithm $\text{Time}_{\text{Coor}}$ for multiple mobile robots path planning with stationary and moving obstacles, we use the data in Table 3 and the information of two stationary obstacles O_1, O_2 and one moving obstacle O_3. For O_1 and O_2, $\alpha_1 = (0.5, 0.6)$, $\alpha_2 = (0.8, 0.9)$, $d_{\text{obs\_1}k} = 0.05$, $d_{\text{obs\_2}k} = 0.03$, $r_{\text{obs}k} = 0$, and $k = 1, 2$. Trajectory of O_3 is one curve from left to right, which is expressed as below and safety distance is set as $d_{\text{obs\_}i3} = 0.16$, $i = 1, 2$:

$$\alpha_{31}(t) = 0.27 + 0.02t + 0.0043t^2,$$
$$\alpha_{32}(t) = 0.6 + 0.01t + 0.001t^2 - 0.00008t^3, \quad t \in [0, 10]. \tag{17}$$

With our approach, for $t_1 \in [0, 5]$ and $t_2 \in [0, 10]$, the expressions of the planned paths are given as below and corresponding curves with time intervals [0, 2.1] [2.1, 4] [4, 10], and [0, 10] are shown in Figure 7. It demonstrates that, with

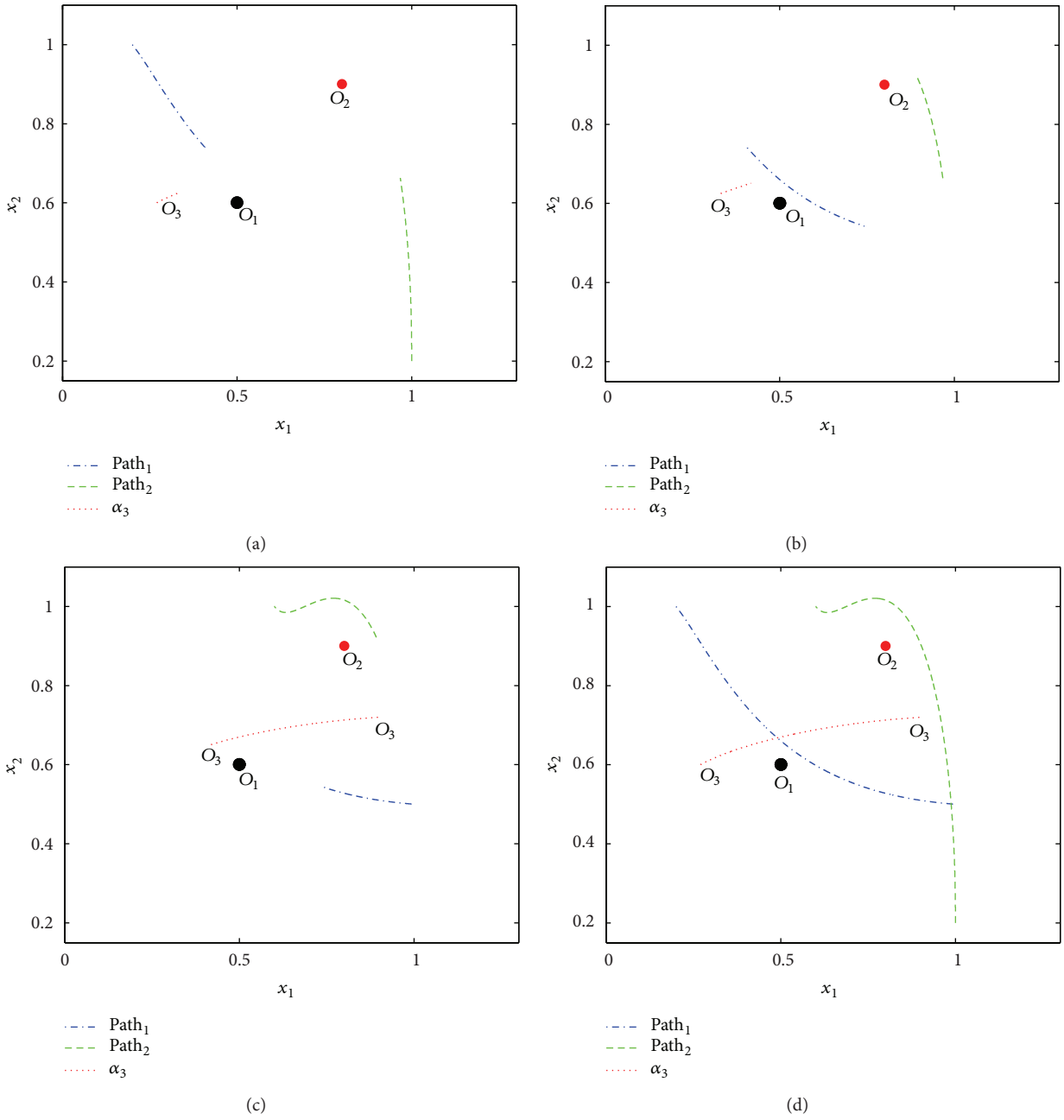


Figure 7: Planned paths of A_1 and A_2 with different time intervals in Section 5.2. (a) In $[0, 2.1]$, (b) in $[2.1, 4]$, (c) in $[4, 10]$, and (d) in $[0, 10]$.

our planned paths, there is no collision among the motion process of the two mobile robots and the three obstacles static and moving:

$$\begin{aligned} x_{11}(t_1) &= 0.2 + 0.0711t_1 + 0.0095t_1^2 + 0.0017t_1^3, \\ x_{21}(t_1) &= 1 - 0.08t_1 - 0.04t_1^2 + 0.01t_1^4 - 0.0006t_1^4, \\ x_{12}(t_2) &= 1 + 0.0011t_2 - 0.009t_2^2 + 0.0005t_2^3, \\ x_{22}(t_2) &= 0.2 + 0.25t_2 - 0.011t_2^2 - 0.003t_2^3 + 0.0002t_2^4. \end{aligned} \tag{18}$$

Table 3: Initial information of mobile robots A_i $(i = 1, 2)$ in Section 5.2.

A_i	P_{i0}	P_{if}	$T_{\max}$	$v_{\max}$	$a_{\max}$
A_1	$[0.2, 1, -\pi/4]^T$	$[1, 0.5, -3\pi/4]^T$	5	0.3	0.4
A_2	$[1, 0.2, -3\pi/4]^T$	$[0.6, 1, -3\pi/4]^T$	10	0.25	0.4

The v-t and a-t curves are given in Figures 8(a) and 8(b), respectively. From Figure 8, it is verified that the velocity and acceleration constraints are satisfied, and then the kinematics constraints are met as well.

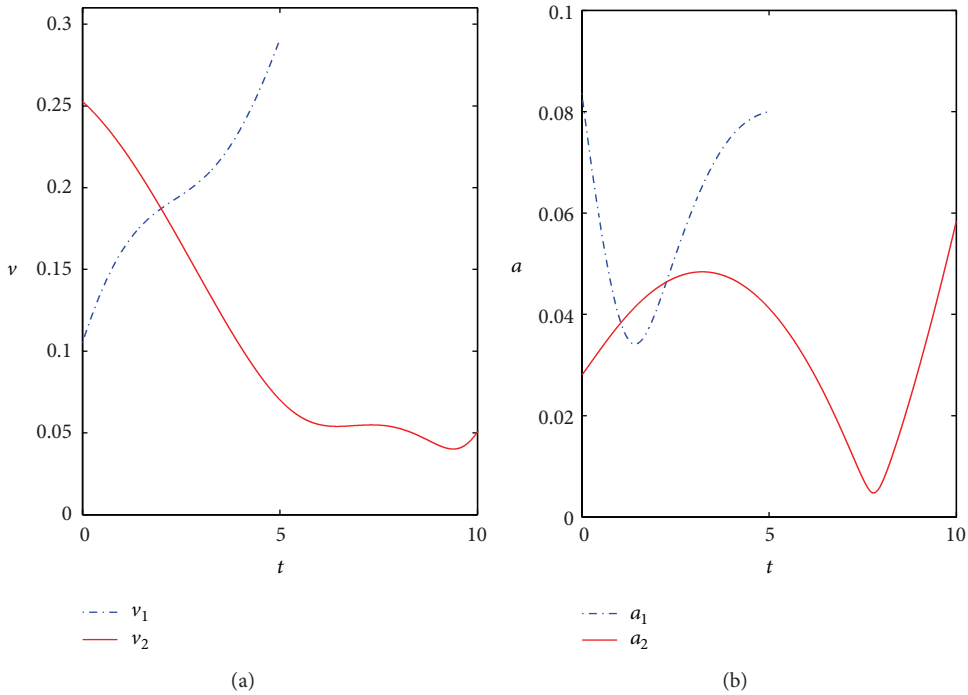


FIGURE 8: (a) v-t, (b) a-t in Section 5.2.

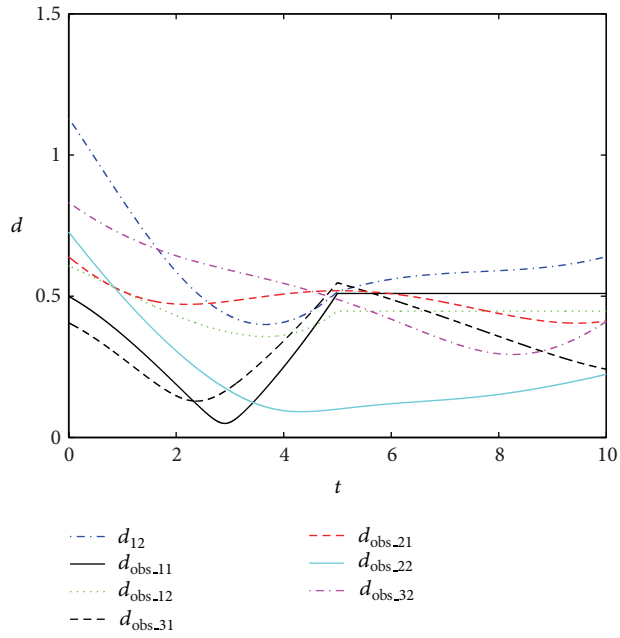


FIGURE 9: Variation curves of safety distance d-t in Section 5.2.

Distances variation curves d_{12}, d_{obs_11}, d_{obs_12}, d_{obs_13}, d_{obs_21}, d_{obs_22}, and d_{obs_23} between mobile robots A_1 and A_2, A_i and O_k ($i = 1, 2$, $k = 1, 2, 3$) are shown in Figure 9. In Figure 9, the safety distance constraints are satisfied as all the distances are in accordance with the corresponding requirements.

With this case, it is verified that our approach is effective for multiple mobile robots path planning under the time-varying environment with stationary and moving obstacles.

6. Discussion

In the case of Section 5.1.1, from Figure 1, it is shown that the initial and final poses of A_1 and A_2 are interchanged with each other. Above situations are common in the path planning problems which belong to the classical deadlock problems. If using one multiphase approach, it is hard to present the concerted ideal solution. However, with our overall conflict resolution and time baseline coordination algorithm, we can dispose this problem perfectly. Meanwhile, in the case of Section 5.1.2, from Figures 4, 5, and 6, we can assert that our approach can handle the path planning problems under the complex time-varying environment. Consequently, those numerical case studies substantiate that our proposed approach would do great deeds for robotic industrial production.

In the case of Section 5.2, from Figure 7, although the trajectory of the moving obstacles almost traverses all the boundaries, no conflict among the planned paths exists. From this case, it is firmly verified that our approach can resolve the path planning problem under the congestive situations with static and moving obstacles which occupied the motion area.

In all the numerical cases, we consider the poses constraints of mobile robots, which have not been achieved in most literatures such as [19]. Thus, our algorithm validates for the multiple mobile robots path planning problems with poses' constraints.

7. Conclusions

The path planning problem for multiple mobile robots under the time-varying environment with stationary and moving obstacles has been studied. The considered objective of our

Input: Admissible error between two consecutive values of path length function: ε, number of the interval $[0, t_{fi}]$ equal parts: F, and coefficient of polynomials: m and n.
Output: Optimal path parameters of A_i: A^i, B^i, T_A^i and T_B^i, path length s_i.
(1) Set $m = 1, n = 1$.
(2) NLPP(ε, F, m, n)
(3) **If** NLPP is infeasible
(4) **then** m++, n++
(5) Transfer to (2)
(6) **else** optimal path parameters A^i, B^i, T_A^i and T_B^i, path length s_i^0
(7) **end if**
(8) m++, n++
(9) NLPP(ε, F, m, n)
(10) **A**ssign the results of (9) to A^i, B^i, T_A^i and T_B^i, s_i^1
(11) **if** abs(s_i^0, s_i^1) > ε
(12) **then** Transfer to (8)
(13) **else** optimal path length value $s_i = s_i^1$
(14) **return** optimal path parameters A^i, B^i, T_A^i and T_B^i, path length s_i

Algorithm 2: $\text{Trap}_{\text{TNLPPs}}(\varepsilon, F, m, n)$.

formulated TNLPPs is to simultaneously minimize all the paths' lengths of mobile robots while subjecting to generous constraints. The algorithm consists of the overall conflict resolution and time baseline coordination is suggested to solve this problem. With overall conflict resolution, all conflicts among the planned paths are removed in our objective function. By using the time baseline coordination method, we can attain the high-quality planned paths. With our approach, all the paths just need to be calculated only once; there is no need to calculate the path for each mobile robot one by one. Furthermore, the phase of conflict coordination is not required any more. Numerical examples under various scenarios are utilized to validate the efficiency of our approach. Moreover, since generous metaheuristics have been proposed and developed in recent years, we can integrate our algorithm with those heuristics methods to optimize the analytical solving process in the near future.

Appendix

See Algorithm 2.

Conflict of Interests

The authors declare that there is no conflict of interests regarding the publication of this paper.

Acknowledgments

The authors wish to sincerely thank the associate editor and anonymous reviewers for their constructive and valuable comments which lead to the better presentation of this paper. The authors are partially supported by the National Science Foundation of China (51309186, 51379170, 61304043, and 71125001) and the Fundamental Research Funds for the Central Universities (WUT 2013-IV-105).

References

[1] E. Garcia, M. Jimenez, P. de Santos, and M. Armada, "The evolution of robotics research," *IEEE Robotics and Automation Magazine*, vol. 14, no. 1, pp. 90–103, 2007.

[2] Y. Ma, H. Wang, and M. Zamirian, "A novel approach for multiple mobile objects path planning: parametrization method and conflict resolution strategy," *Physics Letters A*, vol. 376, no. 4, pp. 377–386, 2012.

[3] Y. Ma, M. Zamirian, Y. Yang, Y. Xu, and J. Zhang, "Path planning for mobile objects in four-dimension based on particle swarm optimization method with penalty function," *Mathematical Problems in Engineering*, vol. 2013, Article ID 613964, 9 pages, 2013.

[4] Z. Li and J. Canny, *Nonholonomic Motion Planning*, vol. 192, Kluwer Academic Publishers, New York, NY, USA, 1993.

[5] S. Rathinam, R. Sengupta, and S. Darbha, "A resource allocation algorithm for multivehicle systems with nonholonomic constraints," *IEEE Transactions on Automation Science and Engineering*, vol. 4, no. 1, pp. 98–104, 2007.

[6] J. Chakraborty, A. Konar, L. Jain, and U. Chakraborty, "Cooperative multi-robot path planning using differential evolution," *Journal of Intelligent and Fuzzy Systems*, vol. 20, no. 1-2, pp. 13–27, 2009.

[7] G. Klančar and I. Škrjanc, "A case study of the collision-avoidance problem based on Bernstein-Bézier path tracking for multiple robots with known constraints," *Journal of Intelligent and Robotic Systems*, vol. 60, no. 2, pp. 317–337, 2010.

[8] I. Skrjanc and G. Klancar, "Optimal cooperative collision avoidance between multiple robots based on Bernstein-Bézier curves," *Robotics and Autonomous Systems*, vol. 58, no. 1, pp. 1–9, 2010.

[9] M. Bennewitz, W. Burgard, and S. Thrun, "Finding and optimizing solvable priority schemes for decoupled path planning techniques for teams of mobile robots," *Robotics and Autonomous Systems*, vol. 41, no. 2-3, pp. 89–99, 2002.

[10] S. Chiddarwar and N. R. Babu, "Conflict free coordinated path planning for multiple robots using a dynamic path modification

sequence," *Robotics and Autonomous Systems*, vol. 59, no. 7-8, pp. 508–518, 2011.

[11] J. van den Berg and M. Overmars, "Prioritized motion planning for multiple robots," in *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS '05)*, pp. 430–435, August 2005.

[12] S. Liu, D. Sun, and C. Zhu, "Coordinated motion planning for multiple mobile robots along designed paths with formation requirement," *IEEE/ASME Transactions on Mechatronics*, vol. 16, no. 6, pp. 1021–1031, 2011.

[13] R. Rashmi, R. Pratyusha, T. Md, S. Swarnalipi, B. Balabantaray, and S. Mohapatra, "Navigational path planning of multi-robot using honey bee mating optimization algorithm (HBMO)," *International Journal of Computer Applications*, vol. 27, no. 11, pp. 1–8, 2011.

[14] H. Wang, Y. Ma, Y. Xie, and M. Guo, "Mobile robot optimal path planning based on smoothing a∗ algorithm," *Journal of Tongji University*, vol. 38, no. 11, pp. 1647–1655, 2010.

[15] T. Berglund, A. Brodnik, H. Jonsson, M. Staffanson, and I. Soderkvist, "Planning smooth and obstacle-avoiding b-spline paths for autonomous mining vehicles," *IEEE Transactions on Automation Science and Engineering*, vol. 7, no. 1, pp. 167–172, 2010.

[16] F. Ghilardelli, G. Lini, and A. Piazzi, "Planning smooth and obstacle-avoiding b-spline paths for autonomous mining vehicles," *IEEE Transactions on Automation Science and Engineering*, vol. 11, no. 1, pp. 187–203, 2014.

[17] J. Yang, Z. Qu, J. Wang, and K. Conrad, "Comparison of optimal solutions to real-time path planning for a mobile vehicle," *IEEE Transactions on Systems, Man, and Cybernetics A: Systems and Humans*, vol. 40, no. 4, pp. 721–731, 2010.

[18] M. Zamirian, A. Kamyad, and M. Farahi, "A novel algorithm for solving optimal path planning problems based on parametrization method and fuzzy aggregation," *Physics Letters A*, vol. 373, no. 38, pp. 3439–3449, 2009.

[19] Y. Wang, D. Lane, and G. Falconer, "Two novel approaches for unmanned underwater vehicle path planning: constrained optimization and semi-infinite constrained optimization," *Robotica*, vol. 18, no. 2, pp. 123–142, 2000.

Hindawi Publishing Corporation
Mathematical Problems in Engineering
Volume 2014, Article ID 526781, 8 pages
http://dx.doi.org/10.1155/2014/526781

Research Article

Construction of Fuzzy Map for Autonomous Mobile Robots Based on Fuzzy Confidence Model

Jung-Fu Hou,[1] Yau-Zen Chang,[1] Ming-Hsi Hsu,[1] Shih-Tseng Lee,[2] and Chieh-Tsai Wu[2]

[1] *Department of Mechanical Engineering, Chang Gung University, Taoyuan 33302, Taiwan*
[2] *Department of Neurosurgery, Chang Gung Memorial Hospital, Taoyuan 33305, Taiwan*

Correspondence should be addressed to Yau-Zen Chang; zen@mail.cgu.edu.tw

Received 15 January 2014; Revised 23 April 2014; Accepted 25 April 2014; Published 16 June 2014

Academic Editor: Leo Chen

This paper presents the use of fuzzy models to explicitly consider sensor uncertainty and finite resolution in solving the SLAM (simultaneous localization and mapping) problem for autonomous mobile robots. The approach establishes fuzzy confidence models in describing occupied obstacles and available space. The problem is transformed into an optimization task of minimizing the alignment error between newly scanned local fuzzy maps and selected parts of a developing global fuzzy map. In aligning local fuzzy maps into a global fuzzy map, we developed a prediction strategy to crop the most potential part from the sensed local fuzzy maps to be overlapped with the global fuzzy map. A mobile vehicle equipped with a laser range finder, the Hokuyo URG-04LX, is used to demonstrate the procedure of fuzzy map building. Experimental results show that the proposed architecture is effective in generating a comprehensive global fuzzy map, which is suitable for both human comprehension and path design during real-time navigation.

1. Introduction

An essential task of an autonomous mobile robot is to determine its location and construct a map of its environment, usually denoted as the work of solving the SLAM (simultaneous localization and mapping) problem [1–3]. Self-localization is about finding the location of a robot in a map, while mapping is about constructing a referable map when the robot is moving in an unknown or changing environment.

Autonomous map construction has been under extensive research for decades [4–9]. For instance, Chong and Kleeman [5] used a sonar sensor and a positioning sensor, and Jaradat and Langari [6] used a sonar sensor in developing the OGM (occupancy grid map) method, where the environment is simplified into occupied and vacant grids. Guivant et al. [7] used encoders in cooperation with a laser range finder for positioning. Davison and Kita [8] combined an accelerometer and two dynamic video cameras to construct irregular maps. Tomono [9] used baselines as the basis for a video camera to choose the characteristic points for map reconstruction.

There are various kinds of sensors developed for these tasks, such as sonar [10, 11], laser range finders [12], and video cameras [13]. Sonar is effective in detecting range, but only a narrow region can be detected at one time. Laser range finders can effectively provide 2D environmental information at high refresh rate, up to 10 frames per sec, but may fail to sense black objects and complex 3D obstacles. Moreover, video cameras can emulate the capability of human eyes, but huge computing power is required for real-time implementation.

An early work [14] proposed a fuzzy model for the sonar sensing, but the paper lacks detailed procedures for the SLAM problem. Inspired by the research, this paper presents the use of a fuzzy model to explicitly consider sensor uncertainty and finite resolution of laser range finders in solving the SLAM problem.

Our proposed system is realized by establishing a fuzzy confidence model, which is composed of sensed obstacles and assured space based on sensor readings. The SLAM problem is transformed into an optimization task of minimizing the alignment error between newly scanned local fuzzy maps and selected parts of a developing global fuzzy map. The task is then solved by the Cuckoo search optimization algorithm [15, 16]. Being a nature-inspired meta-heuristic algorithm,

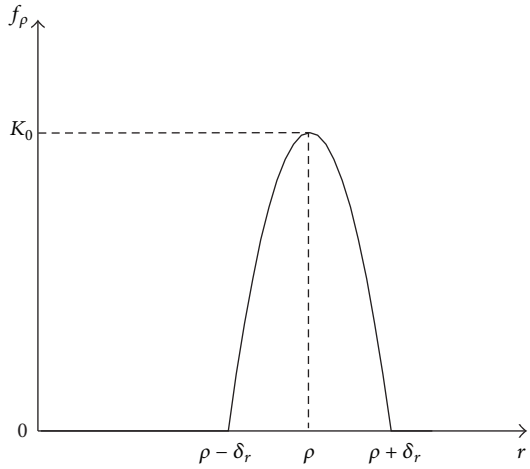


Figure 1: The profile of a confidence function f_ρ which varies around a given range reading ρ.

the algorithm can efficiently provide a set of optimal solutions within a reasonable number of cost function evaluations.

To ease the search, a prediction strategy is applied such that only the most promising portions of the newly scanned local fuzzy maps and last fuzzy maps are involved in the registration. The search starts with adequate knowledge of moving direction, which makes essentially no assumption about the environment and is able to incrementally build a global map in real time. A specific laser range finder, the Hokuyo URG-04LX, is used to demonstrate the use of the confidence model in the building of fuzzy maps.

2. Certainty Model of the Sensor

2.1. Characteristics of the Laser Range Finder. Most distance sensors, such as laser range finders, suffer from both measurement error and finite resolution, which introduce uncertainties deteriorating with distance. Knowledge about these uncertainties may be described explicitly by fuzzy membership functions in order to build confidence models.

To model the knowledge of measurement error in terms of the degree of certainty, confidence level for the existence of an obstacle can be described as a function symmetric to a given range reading ρ. Assuming that K_0 is a parameter corresponding to the maximum confidence level, the confidence function, f_ρ, can be described as [14]

$$f_\rho(r) = \begin{cases} K_0 \cdot \left[1 - \left(\dfrac{r-\rho}{\delta_r}\right)^2\right], & \text{for } |r-\rho| \le \delta_r, \\ 0, & \text{otherwise.} \end{cases} \quad (1)$$

The profile of f_ρ is depicted in Figure 1, where δ_r is the bound of measurement error. According to the figure, the confidence level decreases along a parabolic shaped trajectory toward zero as the distance to the range reading ρ increases.

For a typical distance sensor, the effects of finite lateral resolution can be formulated into another confidence function similar to (1). Hence, we create a hybrid function $B_{\rho,\phi}$ to represent the level of confidence for the combination of measurement error and finite resolution as

$$B_{\rho,\phi}(r,\theta) = \begin{cases} K_{r\theta} \cdot \left[1 - \left(\dfrac{r-\rho}{\delta_r}\right)^2\right] \\ \quad \times \left[1 - \left(\dfrac{\theta-\phi}{\delta_\theta}\right)^2\right], & \text{when } |r-\rho| \le \delta_r, |\theta-\phi| \le \delta_\theta, \\ 0, & \text{otherwise.} \end{cases} \quad (2)$$

As depicted in Figure 2, the profile of the confidence level $B_{\rho,\phi}$ is in the polar coordinate system (r,θ), which is symmetric to both the range reading ρ and angular reading ϕ, where δ_r is the bound of longitudinal distance error and δ_ϕ is the bound of finite lateral resolution. The confidence level decreases toward zero as the distance to the range reading ρ or angular reading ϕ increases.

Information about the existence of obstacles is crucial for the construction for maps, providing valuable environment knowledge to the human supervisor. On the other hand, the information of free space between the sensor and the obstacles is vital for navigation of moving robots, since sensors cannot provide information behind obstacles. The confidence function which describes the availability of free space for navigation is defined as a function $P_{\rho,\phi}(r,\theta)$ corresponding to a set of range reading ρ and angular reading ϕ in the polar coordinate system:

$$P_{\rho,\phi}(r,\theta) = \begin{cases} \left(\dfrac{r-\rho}{\delta_r}\right)^2 \\ \quad \cdot \left[1 - \left(\dfrac{\theta-\varphi}{\delta_\theta}\right)^2\right], & \text{when } \rho - \delta_r \le r \le \rho, |\theta-\phi| \le \delta_\theta, \\ 1 - \left(\dfrac{\theta-\varphi}{\delta_\theta}\right)^2, & \text{when } r < \rho - \delta_r,\ |\theta-\phi| \le \delta_\theta, \\ 0, & \text{otherwise.} \end{cases} \quad (3)$$

The profile of $P_{r,\theta}$ is depicted in Figure 3. We have that the 3D shape is the opposite to that of Figure 2. According to the figure, we have full confidence about the available space between the observer and a measured obstacle located at (ρ,ϕ). On the contrary, we have little confidence about the space behind the sensed obstacle.

2.2. Certainty Model of a Specific Laser Range Finder. The Hokuyo URG-04LX is a popular laser range finder (LRF) which uses a 785 nm semiconductor laser beam. It has a fixed scanning range of 60 to 300° with a 0.36° angular resolution and a 100 msec scan rate. The data transfer rate can be set at 9 Mbps when connected via USB.

The LRF has a quoted range of effective measurement between 20 and 4,095 mm. The measurement error is ±10 mm for distances of less than 1 m. For greater distances, the error is ±1% of the range readings [17, 18]. The variation of

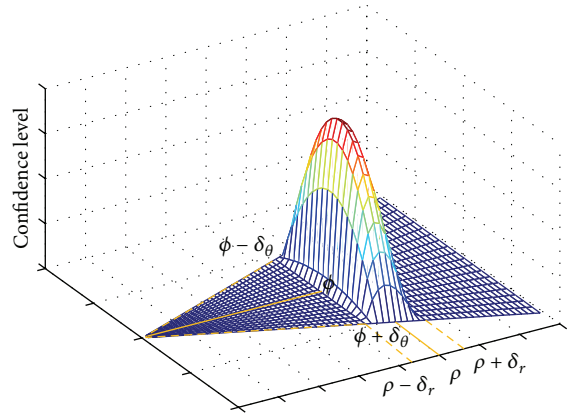


FIGURE 2: The confidence function $B_{\rho,\phi}$ of a distance sensor combines the effects of measurement error and finite resolution around a set of range reading ρ and angular reading ϕ.

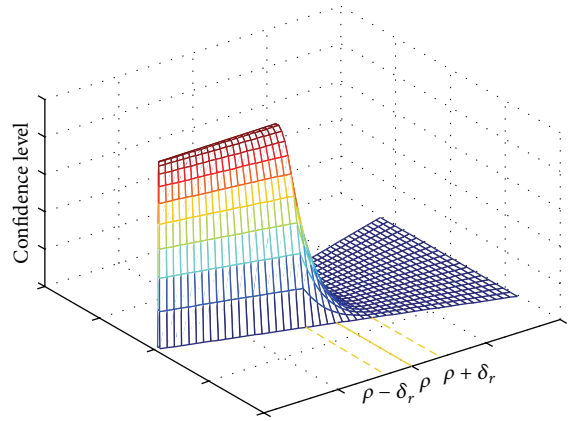


FIGURE 3: The confidence function $P_{r,\theta}$ represents the confidence in the availability of space based on range reading ρ and angular reading ϕ.

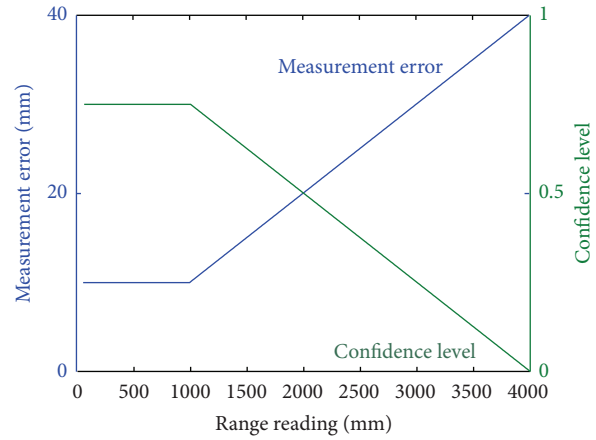


FIGURE 4: Graphic illustration of measurement error as a function of the range reading. The complementary curve is used as the confidence function.

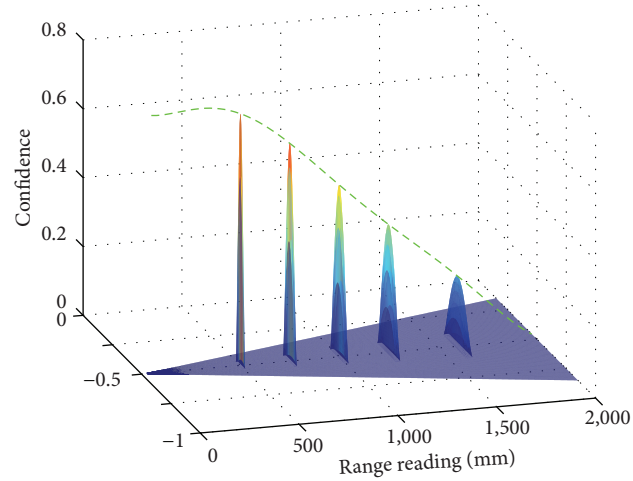


FIGURE 5: Profile of confidence functions corresponding to different range readings.

measurement error in terms of measured distance is depicted in Figure 4. As confidence in measured values decreases with the increase of measurement error, the complementary curve to the measurement error curve is defined as the confidence curve and illustrated in the figure, which can be fitted by a 6th order polynomial function:

$$K_{r\theta}(r) = 4.70 \times 10^{-21} \times r^6 - 5.96 \times 10^{-17} \times r^5 + 2.82 \times 10^{-13} \times r^4 - 5.96 \times 10^{-10} \times r^3 + 4.56 \times 10^{-7} \times r^2 - 1.12 \times 10^{-4} \times r + 0.76. \quad (4)$$

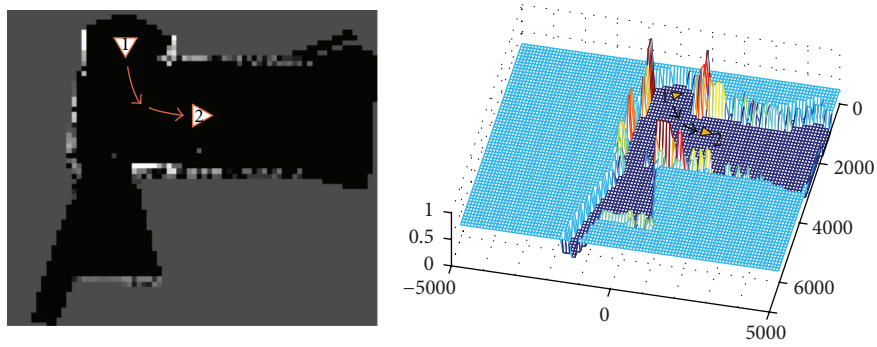
The confidence level, which is a function of range reading as shown in Figure 4, is used as the parameter $K_{r\theta}$ in (2) with δ_θ being 0.36°. In order to further clarify how the confidence levels vary with distance measure, 5 confidence levels with respect to corresponding obstacle detections are demonstrated in Figure 5, where the dotted green curve is generated by the closed form of (4).

2.3. Aggregation for Fuzzy Map Construction. The construction of a local fuzzy map of obstacles at time t, denoted as $M_o(t)$, is based on the aggregation of several sets of range reading and angular reading, (ρ_k, θ_k), in polar coordinates:

$$\mu_{\text{Occupied}} = M_o(t) = \bigcup_k B_{r\theta}(\rho_k(t), \phi_k(t)). \quad (5)$$

Similarly, a local fuzzy map of space at time t, denoted as $M_s(t)$, is based on the aggregation of several sets of range reading and angular reading, (ρ_k, θ_k), in polar coordinates:

$$\mu_{\text{Space}} = M_s(t) = \bigcup_k P_{r\theta}(\rho_k(t), \phi_k(t)). \quad (6)$$



(a) 2D fuzzy map (b) 3D fuzzy map

FIGURE 6: Illustration of fuzzy maps showing both the occupied obstacles and available space. The space is in the darkest color in the diagrams.

Furthermore, to construct a fuzzy map simultaneously suitable for human comprehension and path design, the occupied obstacles and available space should be integrated together:

$$\begin{aligned}\mu_{\text{Map}} &= M_{\text{all}}(t) = \left(0.3 \cup M_o(t)\right) \cap \overline{M_s}(t) \\ &= \min\left(\max\left(0.3, M_o(t)\right), \overline{M_s}(t)\right),\end{aligned} \tag{7}$$

where $\overline{M_s}(t)$ is the complement of $M_s(t)$ and the value 0.3 is selected to distinguish uncertain regions from assured occupied and space regions.

A practical implementation of (7) is demonstrated in Figure 6 where the triangles marked with 1 and 2 are the initial and final positions, respectively, of a moving robot carrying a laser range finder. Figures 6(a) and 6(b) are two-dimensional diagram and three-dimensional diagram, respectively. Note that the space is in the darkest color showing the least level, which is suitable for path planning. The largest areas with gray value, assigned as 0.3 in this example, are the regions behind sensed obstacles and hence unknown to the sensor.

3. Experimental Study

3.1. Registration between Local Fuzzy Maps. The concept of registration is based on the similarities between two local fuzzy maps sensed at different instances. As a robot moves, the scene changes with the emergence and disappearance of objects and boundaries. These extra items on the maps increase the difficulty in the judge of matching between two succeeding local maps, leading to erroneous results. To alleviate the difficulty and enhance robustness and correctness in the search of coordinate transformation for alignment, a prediction algorithm is proposed. The algorithm exploits the knowledge of current move direction to crop local maps from the sensed maps.

Let $M_o(t)|_{(x,y)}$ and $M_o(t+1)|_{(x,y)}$ be the local fuzzy maps sensed at t and $t+1$ which have been transformed into rectangular coordinates from their polar coordinate counterparts, $M_o(t)$ and $M_o(t+1)$, respectively. Besides, let $p_k(t)$ and $p_j(t+1)$ be position vectors with confidence values at $M_o(t)|_{(x,y)}$ and $M_o(t+1)|_{(x,y)}$, respectively. The problem is to find coordinate transformations, ${}^{t+1}_{t}T(\Delta X_t)$ and ${}^{t}_{t+1}T(\Delta X_t)$, such that the fuzzy maps are aligned with each other. Here ΔX_t is a vector of displacement and rotation for the robot to move between t and $t+1$. To begin with the prediction procedure, a rough estimation of ΔX_t, denoted as $\Delta\overline{X_t}$, is generated based on current moving command. Partial local fuzzy maps, $\widehat{M}_o(t+1)|_{(x,y)}$ and $\widehat{M}_o(t)|_{(x,y)}$, that are cropped from $M_o(t+1)$ and $M_o(t)$, are defined as

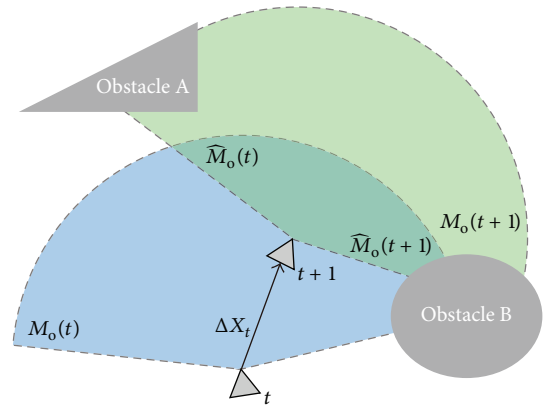


FIGURE 7: The cropping of local maps for registration by the use of estimated motion.

$$\begin{aligned}\widehat{M}_o(t+1)\big|_{(x,y)} = \left\{{}^{t+1}_{t}T\left(\Delta\overline{X_t}\right)p_k(t) \mid p_k(t) \in M_o(t)\big|_{(x,y)}\right\} \\ \cap\, M_o(t+1)\big|_{(x,y)},\end{aligned} \tag{8}$$

$$\begin{aligned}\widehat{M}_o(t)\big|_{(x,y)} = \left\{{}^{t+1}_{t}T\left(\Delta\overline{X_t}\right)p_j(t+1) \mid \right. \\ \left. p_j(t+1) \in M_o(t+1)\big|_{(x,y)}\right\} \cap M_o(t)\big|_{(x,y)}.\end{aligned} \tag{9}$$

The relationship between the local fuzzy maps is illustrated in Figure 7. Besides, the homogeneous transformation

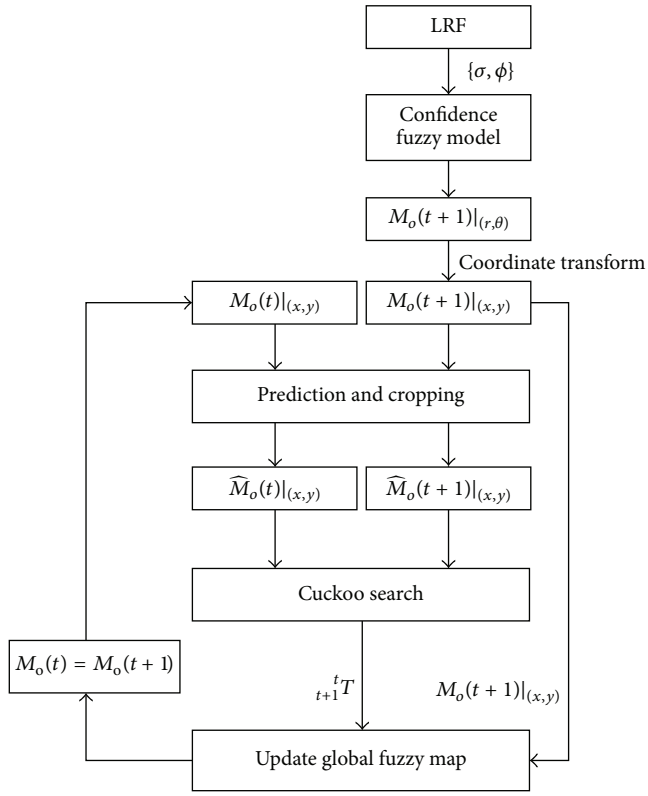


Figure 8: Flowchart of the generation of a global fuzzy map.

matrices, ${}^{t+1}_{t}T(\Delta\overline{X}_t)$ and ${}^{t}_{t+1}T(\Delta\overline{X}_t)$, can be easily obtained by exploiting the relationship ${}^{t+1}_{t}T(\Delta\overline{X}_t) = {}^{t}_{t+1}T(\Delta\overline{X}_t)^{-1} = {}^{t}_{t+1}T(\Delta\overline{X}_t)^{T}$.

3.2. System Flowchart. This section presents an alternative solution to simultaneous localization and mapping (SLAM) problem by direct registration of fuzzy maps of obstacles.

As shown in the flowchart of Figure 8, the data $\{\rho, \phi\}$ sensed by the laser range finder are firstly aggregated into a local fuzzy map $M_o(t+1)$ in polar coordinates and transformed into rectangular coordinate system, denoted as $M_o(t+1)|_{(x,y)}$. The prediction algorithm of (8) and (9) is then used to crop from this local fuzzy map $M_o(t)|_{(x,y)}$ and the last local fuzzy map $M_o(t+1)|_{(x,y)}$ to generate two fuzzy maps, $\widehat{M}_o(t+1)|_{(x,y)}$ and $\widehat{M}_o(t)|_{(x,y)}$, for registration by the Cuckoo search [14], where a cost function of overlap between them is maximized. The registration results in a homogeneous transformation matrix ${}^{t}_{t+1}T$ to be used for the local fuzzy map $M_o(t+1)$ to be integrated into the global fuzzy map in real time. The procedure iterates with the movement of the robot and the receiving of sensed data.

The data $\{\rho, \varphi\}$ sensed by the laser range finder are firstly aggregated into a local fuzzy map $M_o(t+1)$ in polar coordinates and transformed into rectangular coordinate system, denoted as $M_o(t+1)|_{(x,y)}$. The prediction algorithm of (8) and (9) is then used to crop from this local fuzzy map $M_o(t)|_{(x,y)}$ and the last local fuzzy map $M_o(t+1)|_{(x,y)}$ to generate two fuzzy maps, $\widehat{M}_o(t+1)|_{(x,y)}$ and $\widehat{M}_o(t)|_{(x,y)}$, for registration by the Cuckoo search [15], where a cost function of overlap between them is maximized. The registration results in a homogeneous transformation matrix ${}^{t}_{t+1}T$ to be used for the local fuzzy map $M_o(t+1)$ to be integrated into the global fuzzy map in real time. The procedure iterates with the movement of the robot and the receiving of sensed data.

Figure 9 illustrates a situation of two cropped local fuzzy maps, $\widehat{M}_o(t+1)|_{(x,y)}$ and $\widehat{M}_o(t)|_{(x,y)}$, for registration. After successful registration by the Cuckoo search, an optimal transformation matrix ${}^{t}_{t+1}T$ is obtained that has maximum overlapping for these two fuzzy maps, as shown in Figure 10.

4. Experimental Results

A moving robot of three omniwheels (or poly wheels) is built for experimental study; the robot is equipped with a Hokuyo URG-04LX laser range finder and a notebook of Intel Core i5 and 4 GB RAM, as shown in Figure 11.

The proposed procedure was successfully implemented to generate a global fuzzy map shown in Figure 12. In addition to the corridor, a closed passage is selected for robustness evaluation of the approach. The generated global fuzzy map is demonstrated in Figure 13 showing that the proposed method

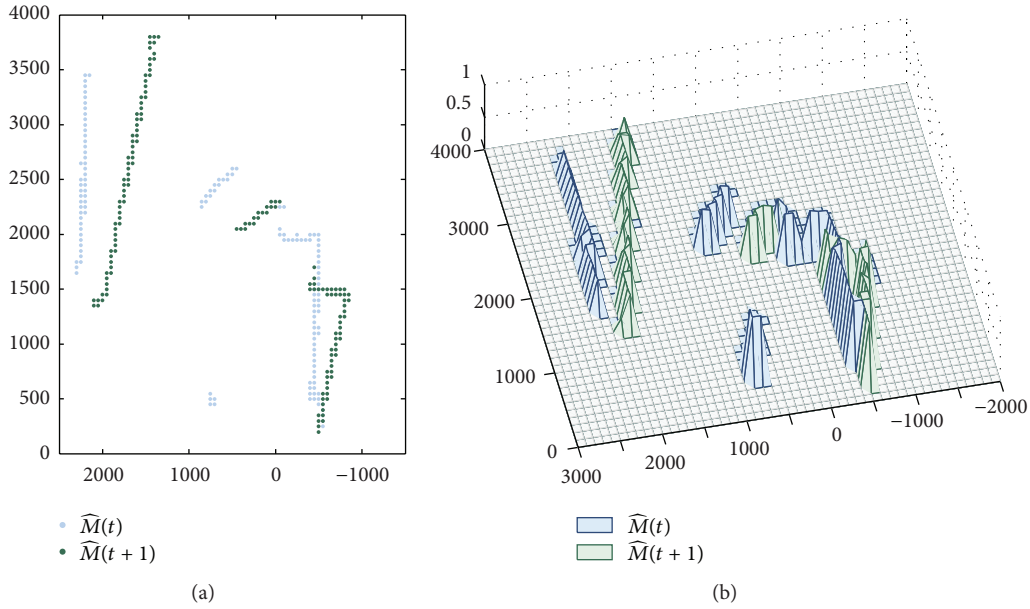


Figure 9: Two local fuzzy maps before registration. (a) 2D Fuzzy maps. (b) 3D fuzzy maps.

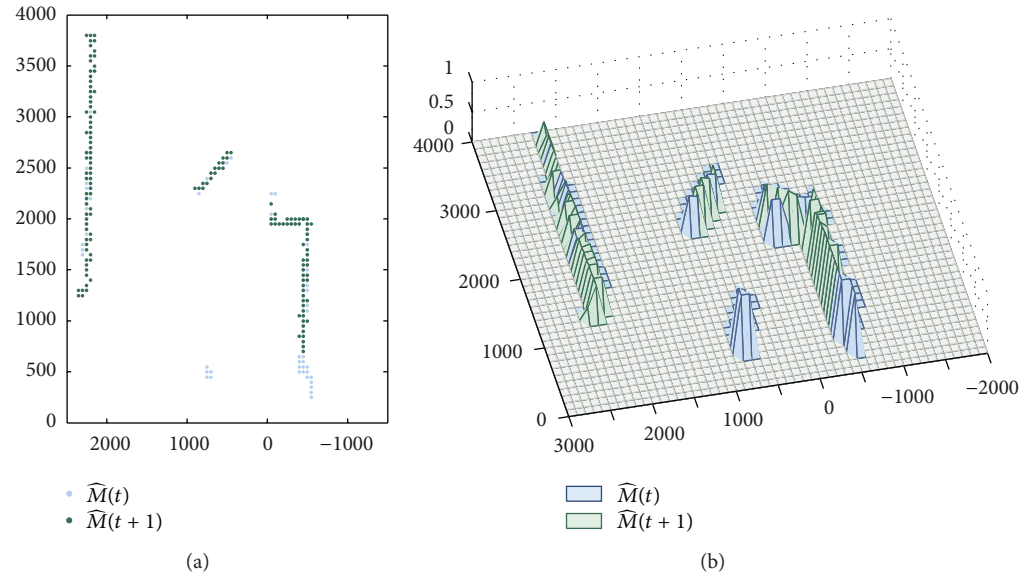


Figure 10: Two local fuzzy maps after registration showing maximum overlapping when compared with Figure 9. (a) 2D fuzzy maps. (b) 3D fuzzy maps.

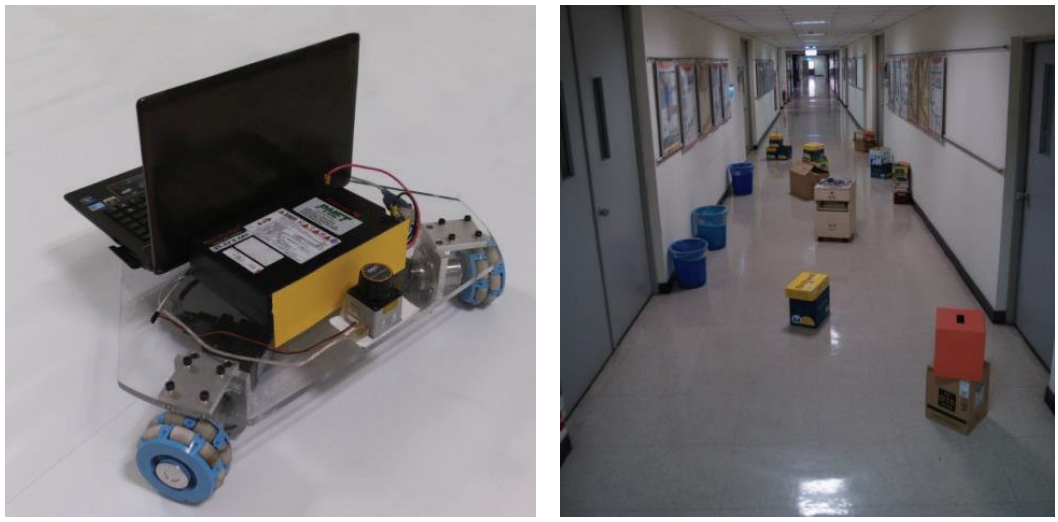
is effective in generating a comprehensible and useful fuzzy map of the environment by the introduction of confidence fuzzy model and the use of the laser range finder.

5. Conclusion

This study proposes an effective SLAM algorithm using fuzzy confidence functions of the laser range finder. The fuzzy confidence functions are functions of both range and anger readings of the sensor which explicitly take sensor uncertainty and finite resolution into consideration. Based on the fuzzy functions, we are able to aggregate sensed data into local fuzzy maps by fuzzy union.

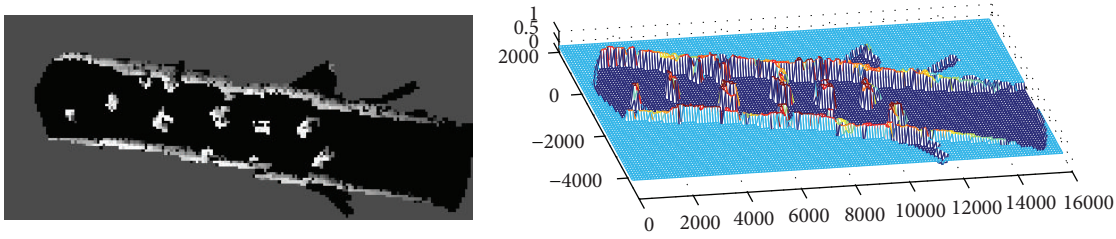
In aligning local fuzzy maps into a global fuzzy map, we developed a prediction strategy to crop the most potential part to be overlapped with global fuzzy map from the sensed local fuzzy maps. This strategy is experimentally evaluated to be effective in finding homogeneous transformation matrices by the Cuckoo search in real time.

In addition to the occupied obstacles, confidence fuzzy functions for available space are also implemented. The integration of occupied obstacles and available space allows us to



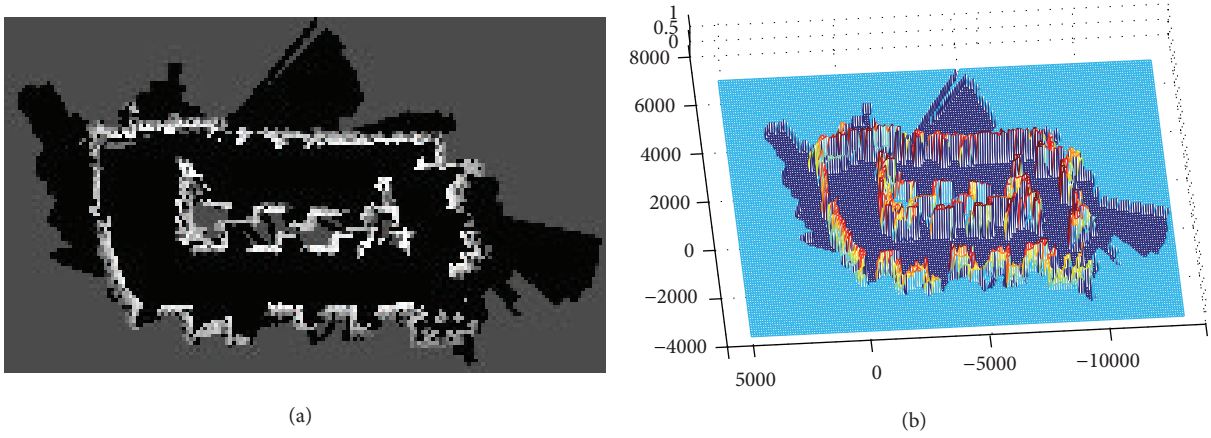
(a) (b)

Figure 11: Experimental setup. (a) A moving robot equipped with three omniwheels. (b) A scenario of a corridor with obstacles along the passage.



(a) (b)

Figure 12: A global fuzzy map generated with the arrangement of Figure 11. (a) 2D fuzzy map. (b) 3D fuzzy map.



(a) (b)

Figure 13: A comprehensive global fuzzy map of a closed passage generated by the proposed approach. (a) 2D fuzzy map. (b) 3D fuzzy map.

generate a comprehensive global fuzzy map that is suitable for both human comprehension and path design. Performance of the proposed architecture is verified by experiment results of a real-time mobile vehicle.

Indeed, fuzzy path planning seems to be more amenable to the proposed fuzzy map than its nonfuzzy counterparts. Further research aimed at exploiting the benefits of fuzzy maps to include measurement errors and uncertainty as

an integral part of a map to improve reliability and robustness is needed to elucidate their role in navigation.

Conflict of Interests

The authors declare that there is no conflict of interests regarding the publication of this paper.

Acknowledgments

The authors are enormously grateful for the support from the National Science Foundation, Taiwan, under Contract nos. NSC 100-2221-E-182-008, NSC 101-2221-E-182-006, and NSC 102-2221-E-182-073; Chang Gung University and Chang Gung Memorial Hospital, Taiwan, under Contract nos. CMRPD1C0021, CMRPD2C0051, and CMRPD2C0052.

References

[1] H. Durrant-Whyte and T. Bailey, "Simultaneous localization and mapping: part I," *IEEE Robotics and Automation Magazine*, vol. 13, no. 2, pp. 99–108, 2006.

[2] T. Bailey and H. Durrant-Whyte, "Simultaneous localization and mapping (SLAM): part II," *IEEE Robotics and Automation Magazine*, vol. 13, no. 3, pp. 108–117, 2006.

[3] T. Sebastian, *Exploring Artificial Intelligence in the New Millennium*, Morgan Kaufmann, San Francisco, Calif, USA, 2003.

[4] S. Thrun, W. Burgard, and D. Fox, *Probabilistic Robotics*, The MIT Press, Cambridge, Mass, USA, 2005.

[5] K. S. Chong and L. Kleeman, "Mobile robot map building from an advanced sonar array and accurate odometry," Technical Report MECSE, 1996.

[6] M. A. K. Jaradat and R. Langari, "Line map construction using a mobile robot with a sonar sensor," in *Proceedings of the IEEE/ASME International Conference on Advanced Intelligent Mechatronics (AIM '05)*, pp. 1251–1256, Monterey, Calif, USA, July 2005.

[7] J. Guivant, E. Nebot, and S. Baiker, "Autonomous navigation and map building using laser range sensors in outdoor applications," *Journal of Robotic System*, vol. 17, no. 10, pp. 565–583, 2000.

[8] A. J. Davison and N. Kita, "3D simultaneous localisation and map-building using active vision for a robot moving on undulating terrain," in *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pp. I384–I391, Kauai, Hawaii, USA, December 2001.

[9] M. Tomono, "3-D localization and mapping using a single camera based on structure-from-motion with automatic baseline selection," in *Proceedings of the IEEE International Conference on Robotics and Automation*, pp. 3342–3347, Barcelona, Spain, April 2005.

[10] S. Guadarrama and A. Ruiz-Mayor, "Approximate robotic mapping from sonar data by modeling perceptions with antonyms," *Information Sciences*, vol. 180, no. 21, pp. 4164–4188, 2010.

[11] S. Noykov and C. Roumenin, "Occupancy grids building by sonar and mobile robot," *Robotics and Autonomous Systems*, vol. 55, no. 2, pp. 162–175, 2007.

[12] A. Diosi and L. Kleeman, "Fast laser scan matching using polar coordinates," *International Journal of Robotics Research*, vol. 26, no. 10, pp. 1125–1153, 2007.

[13] A. J. Davison, I. D. Reid, N. D. Molton, and O. Stasse, "MonoSLAM: real-time single camera SLAM," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 29, no. 6, pp. 1052–1067, 2007.

[14] G. Oriolo, G. Ulivi, and M. Vendittelli, "Fuzzy maps: a new tool for mobile robot perception and planning," *Journal of Robotic Systems*, vol. 14, no. 3, pp. 179–197, 1997.

[15] X.-S. Yang and S. Deb, "Cuckoo search via Lévy flights," in *Proceedings of the World Congress on Nature and Biologically Inspired Computing (NABIC '09)*, pp. 210–214, Coimbatore, India, December 2009.

[16] X.-S. Yang, "Bat algorithm and cuckoo search: a tutorial," *Studies in Computational Intelligence*, vol. 427, pp. 421–434, 2013.

[17] K.-H. Lee and R. Ehsani, "Comparison of two 2D laser scanners for sensing object distances, shapes, and surface patterns," *Computers and Electronics in Agriculture*, vol. 60, no. 2, pp. 250–262, 2008.

[18] R. Bréda, "Experimental measurement of parameters of the spatial scanner Hokuyo URG-04LX," *Przeglad Elektrotechniczny*, vol. 88, no. 5, pp. 132–135, 2012.

Hindawi Publishing Corporation
Mathematical Problems in Engineering
Volume 2014, Article ID 519524, 6 pages
http://dx.doi.org/10.1155/2014/519524

Research Article

Leader-Based Consensus of Heterogeneous Nonlinear Multiagent Systems

Tairen Sun,[1,2] Yongping Pan,[3] and Haoyong Yu[3]

[1] *School of Electrical and Information Engineering, Jiangsu University, Zhenjiang 212013, China*
[2] *Key Laboratory of Autonomous Systems and Networked Control, Ministry of Education, South China University of Technology, Guangzhou 510640, China*
[3] *Department of Biomedical Engineering, National University of Singapore, Singapore 117575*

Correspondence should be addressed to Tairen Sun; suntren@gmail.com

Received 17 March 2014; Revised 24 May 2014; Accepted 25 May 2014; Published 16 June 2014

Academic Editor: Leo Chen

This paper considers the leader-based consensus of heterogeneous multiple agents with nonlinear uncertain systems. Based on the information obtained from the following agents' neighbors, leader observers are designed by the following agents to estimate the leader's states and nonlinear dynamics. Then, to achieve leader-based consensus, adaptive distributed controllers are designed for the following agents to track the designed corresponding leader observers. The effectiveness of the leader observers and distributed consensus controllers are illustrated by formal proof and simulation results.

1. Introduction

In the past decades, cooperative control problems of multiagent have attracted more and more attention from researchers for its wide applications in many areas, such as formation control of mobile robots [1], monitoring [2], medical rescue [3], and environmental boundary tracking [4]. Consensus is a fundamental and important problem in the area of multiagent systems where a networked interaction protocol needs to be designed such that the final states of all agents converge to a common value.

Compared with the leaderless consensus in which the agents' states converge to a common constant value [5, 6], the states of multiagent can converge to a leader's dynamic state in leader-based consensus [7]. Though many results on leader-based multiagent consensus have been obtained [7–14], most of the results considered agents with integrator or linear dynamics. However, most real agent systems are inherently nonlinear and uncertainty terms may exist in the systems. The consensus control in integrator or linear dynamics cannot be used directly in consensus of nonlinear agents. Recently, results [15–18] considered the leader-based consensus of homogeneous nonlinear agents with the assumption that the agents' nonlinear function satisfies the Lipschitz or like-Lipchitz condition. The dynamical differences among agents are neglected in [15–18]. In the paper, we consider the leader-based consensus of heterogenous agents with different nonlinear dynamics.

NNs and fuzzy techniques are wildly used in control design for nonlinear uncertain systems, for their abilities in nonlinear function approximations [19, 20]. In [21–25], the leader-based consensus controllers were designed for heterogeneous nonlinear uncertain multiagent systems. In the consensus control design, the leader's nonlinear dynamics was considered as a bounded disturbance and neural network- (NN-) based consensus control was designed for the following agents.

Distributed estimation via observer design for multiagent consensus is an important topic in the study of multiagent networks, with wide applications especially in sensor networks and robot networks, among many others [26]. Up to now, the results [26–31] have been obtained on multiagent consensus based on the distributed estimation via observer design. However, the considered agents in [26–31] are with

integrator or linear dynamics. In this paper, the considered agents are with nonlinear uncertain dynamics and distributed NN-based leader observers are designed to estimate the leader's state and nonlinear dynamics. Then, to obtain the leader-based consensus, NN controllers are designed for the following agents to track the corresponding leader observers.

2. Preliminaries

2.1. Graph Theory and Some Notations. In this paper, we consider a networked system consisting of N agents $v_1, v_2, \ldots, v_N$ and one leader v_0. The leader's system model and motion are independent of the other agents. Denote $G = (V, E)$ as a general digraph with the nonempty finite set of N nodes or agents $V = \{v_1, v_2, \ldots, v_N\}$ and a set of edges or arcs $E \subseteq V \times V$. Denote the connectivity matrix as $A = [a_{ij}]$ with $a_{ij} > 0$ if $(v_j, v_i) \in E$ and $a_{ij} = 0$ otherwise. Note $a_{ii} = 0$. The set of neighbors of a node v_i is $N_i = \{v_j : (v_j, v_i) \in E\}$, that is, the set of nodes with arcs incoming to v_i. Define the in-degree matrix as a diagonal matrix $D = \mathrm{diag}\{d_1, d_2, \ldots, d_N\}$ with $d_i = \sum_{j \in N_i} a_{ij}$ for node i. Define the Laplacian matrix associated with graph G as $L = D - A$.

Define an augmented graph as $\overline{G} = \{\overline{V}, \overline{E}\}$, where $\overline{V} = \{v_0, v_1, \ldots, v_N\}$ and $\overline{E} \subseteq \overline{V} \times \overline{V}$. Denote the diagonal matrix $B = \mathrm{diag}\{b_1, b_2, \ldots, b_N\}$ as the leader adjacency matrix with $b_i = 0$ if the ith agent is a neighbor of the leader and $b_i = 0$ otherwise.

Lemma 1 (see [22–24]). *Let L be irreducible and B has at least one diagonal entry $b_i > 0$. Then $L + B$ is a nonsingular M-matrix. Define $q = [q_1, \ldots, q_N]^T = (L+B)^{-1}\underline{1}$, $P = \mathrm{diag}\{p_i\} = \mathrm{diag}\{1/q_i\}$, $Q = P(L + B) + (L + B)^T P$. Then P and Q are positive definite matrices.*

2.2. Problem Statement. In the considered leader-based consensus, the leader's nonlinear dynamics is described as follows:

$$\dot{x}_0 = f_0(x_0) \tag{1}$$

with $x_0 \in R$ the state and $f_0(x_0)$ a piecewise continuous function of x_0.

Consider N ($N \geq 2$) followed heterogenous agents with nonlinear uncertain dynamics described as follows:

$$\dot{x}_i = f_i(x_i) + u_i, \quad i = 1, \ldots, N, \tag{2}$$

where $x_i(t) \in R$, u_i denote the state and the control input of the ith agent, respectively; $f_i(x_i)$ is a nonlinear uncertain function, which is assumed to be continuous.

In this paper, we make the following assumptions: the following agent i, $i = 1, \ldots, N$, can obtain its neighbors' states x_j, $j \in N_i$, and the leader's state estimation $\widehat{x}_{0j}$, only some agents can obtain the leader's states x_0, and all the following agents do not know the leader's nonlinear dynamics.

The objective of this paper is to design controllers u_i, $i = 1, \ldots, N$, such that the consensus errors $|x_i - x_0|$ converge to a small neighborhood of zero.

3. Leader-Based Consensus of Multiple Agents

Since the nonlinear functions $f_i(x_i)$, $f_0(x_0)$ are piecewise continuous and unknown to agent i, they can be estimated by agent i using radial basis function (RBF) NNs on a compact set $\Omega_i \in R$ as follows:

$$\widehat{f}_i = W_i^T \phi_i(x_i), \tag{3}$$

$$\widehat{f}_{0i} = W_{0i}^T \varphi_i(\widehat{x}_{0i}). \tag{4}$$

By ideal estimation the $f_i(x_i)$ and $f_0(x_0)$ can be written as

$$f_i(x_i) = W_i^{*T} \phi_i(x_i) + \epsilon_i(x_i), \quad |\epsilon_i(x_i)| \leq \epsilon_{iM}, \tag{5}$$

$$f_0(x_0) = W_{0i}^{*T} \varphi_i(x_0) + \epsilon_{0i}(x_0), \quad |\epsilon_{0i}(x_0)| \leq \bar{\epsilon}_{iM}, \tag{6}$$

where $\phi_i(x_i) \in R^{n_i}$ and $\varphi_i(x_0) \in R^{m_i}$ are suitable basis sets of n_i functions and m_i functions, respectively; W_i and W_{0i} are current estimation of ideal approximation weights W_i^* and W_{0i}^*, respectively; ξ_i and ζ_i are bounds of ideal estimation errors $\epsilon_i(x_i)$ and $\epsilon_0(x_0)$, respectively; and $\widehat{x}_{0i}$ is the estimation of x_0.

Remark 2. According to the definition of ϕ_i, φ_i, W_i, W_{0i}, it is easy to observe that there exist positive numbers W_M, W_{0M}, ϵ_M, and ϵ_{0M}, such that $\|W_i^*\| \leq W_M$, $\|W_{0i}^*\| \leq W_{0M}$, $|\epsilon_i| \leq \epsilon_M$, $|\epsilon_{0i}| \leq \epsilon_{0M}$, $i = 1, 2, \ldots, N$.

3.1. Observers for the Leader Agent. For agent i, the following NN-based observer is designed to estimate the leader's state and nonlinear dynamics:

$$\dot{\widehat{x}}_{0i} = \widehat{f}_{0i} - c \sum_{j \in N_i} a_{ij} \left(\widehat{x}_{0i} - \widehat{x}_{0j}\right) - c b_i \left(\widehat{x}_{0i} - x_0\right). \tag{7}$$

Denote $\underline{x}_0 = [x_0, x_0, \ldots, x_0]^T \in R^N$, $\underline{\widehat{x}}_0 = [\widehat{x}_{01}, \ldots, \widehat{x}_{0N}]^T \in R^N$, $e_0 = \underline{\widehat{x}}_0 - \underline{x}_0$, $e_{0i} = \widehat{x}_{0i} - x_0$. Taking time derivative of e_0 leads to

$$\dot{e}_0 = \widehat{f}_0 - \underline{f}(x_0) - c(L + B) e_0 \tag{8}$$

with $\widehat{f}_0 = [\widehat{f}_{01}, \widehat{f}_{02}, \ldots, \widehat{f}_{0N}]^T \in R^N$, $\underline{f}(x_0) = [f_0(x_0), \ldots, f_0(x_0)]^T \in R^N$.

From (4) and (6), $\widehat{f}_{0i} - f_0(x_0)$ can be expressed as

$$\begin{aligned} \widehat{f}_{0i} - f_0(x_0) &= W_{0i}^T \varphi_i(\widehat{x}_{0i}) - W_{0i}^{*T} \varphi_i(x_0) - \epsilon_{0i} \\ &= -W_{0i}^{*T} \widetilde{\varphi}_{0i} - \widetilde{W}_{0i}^T \varphi_i(\widehat{x}_{0i}) - \epsilon, \end{aligned} \tag{9}$$

where $\widetilde{W}_{0i} = W_{0i}^* - W_{0i}$.

Remark 3. From Remark 2, we know that $-W_{0i}^{*T} \widetilde{\varphi}_{0i} - \epsilon_{0i}$, $i = 1, 2, \ldots, N$, are bounded. Then there exists a positive constant ξ, such that $|-W_{0i}^{*T} \widetilde{\varphi}_{0i} - \epsilon_{0i}| \leq \xi$.

The following theorem illustrates the efficiency of the designed NN-observer.

Theorem 4. *Consider the leader and its following agents with the dynamics described by* (1) *and* (2). *If the NN weights in the observer* (7) *are updated by*

$$\dot{\widehat{W}}_{0i} = -\dot{\widetilde{W}}_{0i} = -F_i^{-1} e_i p_i \varphi_i\left(\widehat{x}_{0i}\right) - k F_i^{-1} \widehat{W}_{0i} \tag{10}$$

then the estimation errors $e_{0i}, \widetilde{W}_{0i}$, $i = 1, 2, \ldots, N$, *are uniformly ultimately bounded. Furthermore, the estimation error* e_{0i} *can be made arbitrarily small by a judicious choice of corresponding gains.*

Proof. Consider the following Lyapunov function: $V_0 = (1/2)e_0^T P e_0 + (1/2)\sum_{i=1}^{N} \widetilde{W}_{0i}^T F_i \widetilde{W}_{0i}$. Take time derivative of V_0 and substitute (8) and (10). Consider

$$\begin{aligned}
\dot{V}_0 &= -c e_0^T P (L+B) e_0 + e_0^T P\left(\hat{f}_0 - \underline{f}\left(x_0\right)\right) + \sum_{i=1}^{N} \widetilde{W}_{0i}^T F_i \dot{\widetilde{W}}_{0i} \\
&= -c e_0^T \left[P(L+B) + (L+B)^T P\right] e_0 \\
&\quad + \sum_{i=1}^{N} p_i e_{0i}\left(-W_{0i}^{*T} \widetilde{\varphi}_{0i} - \epsilon_{0i}\right) \\
&\quad + \sum_{i=1}^{N} \widetilde{W}_{0i}^T \left[F_i \dot{\widetilde{W}}_{0i} - e_i p_i \varphi_i\left(\widehat{x}_{0i}\right)\right] \\
&\le -c e_0^T Q e_0 + \xi \sum_{i=1}^{N} p_i \left|e_{0i}\right| - k \sum_{i=1}^{N} \widetilde{W}_{0i}^T \widetilde{W}_{0i} \\
&\quad + k \sum_{i=1}^{N} \left\|\widetilde{W}_i\right\| W_{0M}.
\end{aligned} \tag{11}$$

Since $p_i|e_{0i}| \le p_{\max}|e_{0i}| \le p_{\max}((e_{0i}^2/2\xi_1) + (\xi_1/2))$ with $p_{\max} = \max\{p_i, i = 1, \ldots, N\}$ and $\|\widetilde{W}_i\| \le (1/2\xi_2)\widetilde{W}_i^T \widetilde{W}_i + (\xi_2/2)$ hold for arbitrary positive numbers ξ_1, ξ_2,

$$\begin{aligned}
\dot{V}_0 \le\ & -\left(c\lambda_{\min}(Q) - \frac{\xi p_{\max}}{(2\xi_1)}\right) e_0^T e_0 - \left(k - \frac{k W_{0M}}{(2\xi_2)}\right) \sum_{i=1}^{N} \widetilde{W}_i^T \widetilde{W}_i \\
&+ 0.5N\left(p_{\max}\xi_1 \xi + \xi_2 W_{0M}\right).
\end{aligned} \tag{12}$$

From (12) we know that the derivative of V_0 is guaranteed to be less than zero, as long as $c\lambda_{\min}(Q) - \xi p_{\max}/(2\xi_1) > 0$, $k - kW_{0M}/(2\xi_2) > 0$, and one of the following conditions holds:

$$\begin{aligned}
\left\|e_0\right\| &\ge \sqrt{\frac{0.5N\left(p_{\max}\xi_1\xi + \xi_2 W_{0M}\right)}{c\lambda_{\min}(Q) - \xi p_{\max}/(2\xi_1)}}, \\
\sum_{i=1}^{N} \widetilde{W}_{0i}^T \widetilde{W}_{0i} &\ge \frac{0.5N\left(p_{\max}\xi_1\xi + \xi_2 W_{0M}\right)}{k - kW_{0M}/(2\xi_2)}.
\end{aligned} \tag{13}$$

Therefore $\dot{V}_0$ is negative outside a compact set. According to the standard Lyapunov theory extension, the estimation errors e_{0i}, $\widetilde{W}_{0i}$, $i = 1, 2, \ldots, N$, are uniformly ultimately bounded. Furthermore, the estimation error e_{0i} can be made arbitrarily small by a judicious choice of corresponding gains. □

3.2. Leader-Based Consensus Control Design. Since the effectiveness of the leader observers's estimation of the leader's state and dynamics, the leader-based consensus problem can be solved if some controllers are designed for the following agents to track the designed observers. In this part, NN-based controllers are designed for the agents i, $i = 1, 2, \ldots, N$, to track the corresponding observers.

Denote $e_i = x_i - \widehat{x}_{0i}$. Take time derivative of e_i and substitute (2) and (8). Consider

$$\dot{e}_i = f_i\left(x_i\right) + u_i - \hat{f}_{0i} + c\sum_{j\in N_i} a_{ij}\left(\widehat{x}_{0i} - \widehat{x}_{0j}\right) + c b_i\left(\widehat{x}_{0i} - x_0\right). \tag{14}$$

If the control law u_i is designed as

$$u_i = -l_i e_i - \hat{f}_i + \hat{f}_{0i} - c\sum_{j\in N_i} a_{ij}\left(\widehat{x}_{0i} - \widehat{x}_{0j}\right) - c b_i\left(\widehat{x}_{0i} - x_0\right), \tag{15}$$

then we have

$$\begin{aligned}
\dot{e}_i &= -l_i e_i + W_i^{*T}\phi_i\left(x_i\right) - W_i^T \phi_i\left(x_i\right) + \epsilon_i \\
&= -l_i e_i + \widetilde{W}_i^T \phi\left(x_i\right) + \epsilon_i.
\end{aligned} \tag{16}$$

Based on the above analysis, we have the following result.

Theorem 5. *Consider the distributed system* (1) *and the leader* (2). *If the control law for the following agents is designed as* (15) *and the NN weights* W_i *are updated as*

$$\dot{W}_i = -\dot{\widetilde{W}}_i = \overline{F}_i^{-1}\phi_i\left(x_i\right) e_i, \tag{17}$$

then tracking errors e_i, $i = 1, 2, \ldots, N$, *are uniformly ultimately bounded and can be made arbitrarily small by appropriate choice of corresponding gains; that is, the leader-based consensus is achieved.*

Proof. Consider the candidate Lyapunov function $V_1 = (1/2)e_i^2 + (1/2)\widetilde{W}_i^T \overline{F}_i \widetilde{W}_i$. Take time derivative of V_1 and substitute (16)-(17). Consider

$$\begin{aligned}
\dot{V}_1 &= e_i\left(-l_i e_i + \widetilde{W}_i^T \phi\left(x_i\right) + \epsilon_i\right) + \widetilde{W}_i^T \overline{F}_i \dot{\widetilde{W}}_i \\
&= -l_i e_i^2 + e_i \epsilon_i + \widetilde{W}_i^T\left(\overline{F}_i \dot{\widetilde{W}}_i + \phi_i\left(x_i\right) e_i\right) \\
&\le -l_i e_i^2 + \left|e_i\right| \epsilon_N.
\end{aligned} \tag{18}$$

Since $|e_i|\epsilon_N \le (e_i^2/2) + (\epsilon_N^2/2)$ holds,

$$\dot{V}_1 \le -\left(l_i - \frac{1}{2}\right) e_i^2 + \frac{\epsilon_N^2}{2}. \tag{19}$$

From (19) we know that $\dot{V}_1$ is negative, as long as $l_i - (1/2) > 0$ and $|e_i| \ge (\epsilon_N/\sqrt{2l_i - 1})$. Therefore, according to the standard Lyapunov theory extension, the tracking errors e_i are uniformly ultimately bounded and can be made arbitrarily small by a judicious choice of corresponding gains.

Since $x_i - x_0 = x_i - \hat{x}_{0i} + \hat{x}_{0i} - x_0$, $i = 1, 2, \ldots, N$, from Theorem 4 and the uniformly ultimately boundedness of $e_i = x_i - \hat{x}_{0i}$, $i = 1, 2, \ldots, N$, we can conclude that the consensus errors $x_i - x_0$, $i = 1, 2, \ldots, N$, are uniformly bounded and can be made arbitrarily small by appropriate choice of corresponding gains. □

4. Simulation Results

Consider a group of nonlinear uncertain agents composed of a leader agent 0 and four following agents 1, 2, 3, 4 described in Figure 1. Let the dynamics of the agents be

$$\begin{aligned} \dot{x}_0 &= \sin x_0 \\ \dot{x}_1 &= \cos x_1 + u_1 \\ \dot{x}_2 &= 2x_2 + \sin x_2 + u_2 \\ \dot{x}_4 &= \cos x_4 + u_4. \end{aligned} \tag{20}$$

Simulations are carried out on Pentinum(R) Dual-Core CPU and Matlab 2008b environments. In the simulation, the initial values of the five agents are $x_0(0) = 1$, $x_1(0) = 2$, $x_2(0) = 4$, $x_3(0) = 3$, and $x_4(0) = 1$ and the NN weights and the observers are initialized to be zero vectors or zero. Choose the design parameters as $k = 3$, $F_i^{-1} = 100$, $c = 5$, $l_1 = 5$, $l_2 = 8$, $l_3 = 9$, and $l_4 = 5$. The simulation results are present in Figures 2–4, where leader-based consensus errors $x_0 - x_i$, $i = 1, 2, 3, 4$, are presented in Figure 2, and the estimation errors $x_0 - \hat{x}_{0i}$, $i = 1, 2, 3, 4$, in the leader observers are presented in Figures 3 and 4 describing the tracking errors $x_i - x_{0i}$, $i = 1, 2, 3, 4$, between the following agents and their corresponding observers. From the simulation results in Figures 2–4, the errors $x_0 - x_i$, $x_0 - \hat{x}_{0i}$, and $x_i - x_{0i}$, $i = 1, 2, 3, 4$, converge to very small neighborhoods of zero after 3 seconds. So, we can conclude that the leader observers (7) are effective to estimate the leader's states and nonlinear dynamics and the leader-based consensus of multiple nonlinear uncertain agents can be achieved under the controllers (15).

5. Conclusions

This paper addressed the NN-observer-based leader-following consensus of heterogeneous multiagent systems with nonlinear uncertain dynamics. NN-based leader observers were designed to estimate the leader's state and nonlinear dynamics. Then NN-based controllers were designed for the following agents to track the corresponding leader observer so that leader-based consensus can be achieved. The effectiveness of the consensus construction method was illustrated by theoretical analysis and simulation results.

In this paper, the considered leader was with nonlinear time-invariant system. The observer-based leader-following consensus of nonlinear uncertain systems, in which the leader is with nonlinear time-varying system, needs to be considered in the future.

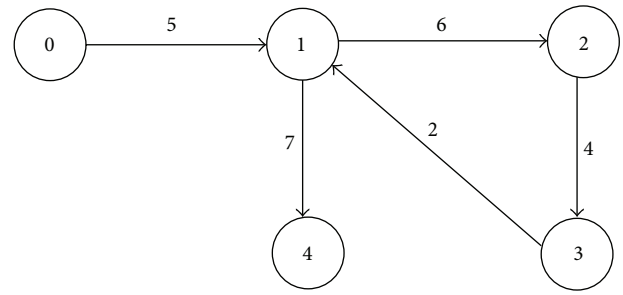


Figure 1: Communication topology of five agents.

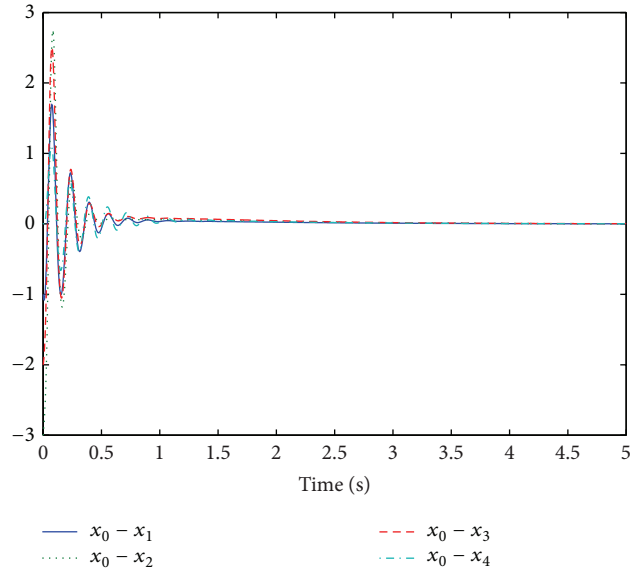


Figure 2: Leader-based consensus errors $x_0 - x_i$, $i = 1, 2, 3, 4$.

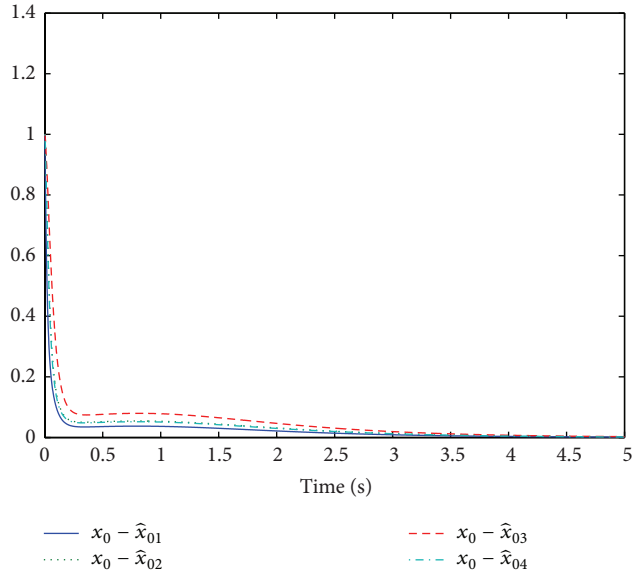


Figure 3: Leader observers' estimation errors $x_0 - \hat{x}_{0i}$, $i = 1, 2, 3, 4$.

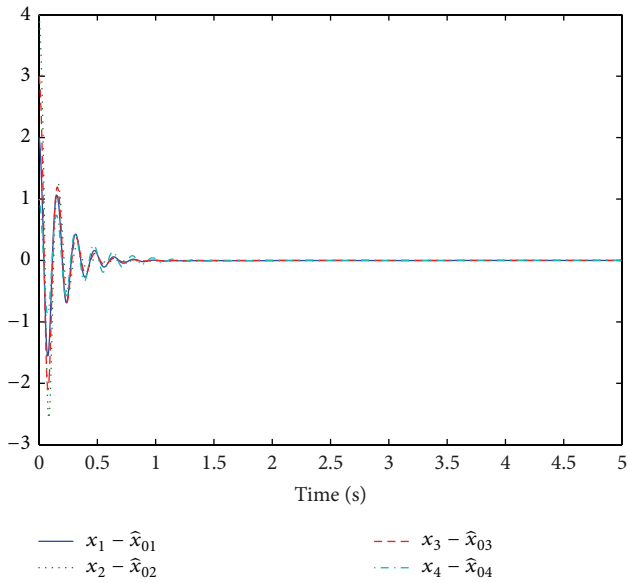


FIGURE 4: The tracking errors $x_i - x_{0i}$, $i = 1, 2, 3, 4$, between the following agents and their corresponding observers.

Conflict of Interests

The authors declare that there is no conflict of interests regarding the publication of this paper.

Acknowledgments

This work was supported by the National Natural Science Foundation of China (61304073), the Natural Science Foundation of Jiangsu Province (BK20130536 and BK20130533), China Postdoctoral Science Foundation (2013M541615 and 2013M540421), Postdoctoral Science Foundation of Jiangsu Province (1301044B), and the Priority Academic Program Development of Jiangsu Higher Education Institutions.

References

[1] T. Sun, F. Liu, H. Pei, and Y. He, "Observer-based adaptive leader-following formation control for non-holonomic mobile robots," *IET Control Theory & Applications*, vol. 6, no. 18, pp. 2835–2841, 2012.

[2] J. T. Feddema, C. Lewis, and D. A. Schoenwald, "Decentralized control of cooperative robotic vehicles: theory and application," *IEEE Transactions on Robotics and Automation*, vol. 18, no. 5, pp. 852–864, 2002.

[3] J. S. Jennings, G. Whelan, and W. F. Evans, "Cooperative search and rescue with a team of mobile robots," in *Proceedings of the 8th International Conference on Advanced Robotics (ICAR '97)*, pp. 193–200, July 1997.

[4] T. Sun, H. Pei, Y. Pan, and C. Zhang, "Robust adaptive neutral network control for environmental boundary tracking by mobile robots," *International Journal of Robust and Nonlinear Control*, vol. 23, no. 2, pp. 123–136, 2013.

[5] Z.-G. Hou, L. Cheng, and M. Tan, "Decentralized robust adaptive control for the multiagent system consensus problem using neural networks," *IEEE Transactions on Systems, Man, and Cybernetics, Part B*, vol. 39, no. 3, pp. 636–647, 2009.

[6] Z.-X. Liu and Z.-Q. Chen, "Discarded consensus of network of agents with state constraint," *IEEE Transactions on Automatic Control*, vol. 57, no. 11, pp. 2869–2874, 2012.

[7] L. Cheng, Y. Wang, Z.-G. Hou, M. Tan, and Z. Cao, "Sampled-data based average consensus of second-order integral multi-agent systems: switching topologies and communication noises," *Automatica*, vol. 49, no. 5, pp. 1458–1464, 2013.

[8] S. Li, H. Du, and X. Lin, "Finite-time consensus algorithm for multi-agent systems with double-integrator dynamics," *Automatica*, vol. 47, no. 8, pp. 1706–1712, 2011.

[9] F. Jiang, L. Wang, and G. Xie, "Consensus of high-order dynamic multi-agent systems with switching topology and time-varying delays," *Journal of Control Theory and Applications*, vol. 8, no. 1, pp. 52–60, 2010.

[10] K. Liu, G. Xie, and L. Wang, "Consensus for multi-agent systems under double integrator dynamics with time-varying communication delays," *International Journal of Robust and Nonlinear Control*, vol. 22, no. 17, pp. 1881–1898, 2012.

[11] F. Xiao and T. Chen, "Sampled-data consensus for multiple double integrators with arbitrary sampling," *IEEE Transactions on Automatic Control*, vol. 57, no. 12, pp. 3230–3235, 2012.

[12] Y. Zhang and Y.-P. Tian, "Allowable sampling period for consensus control of multiple general linear dynamical agents in random networks," *International Journal of Control*, vol. 83, no. 11, pp. 2368–2377, 2010.

[13] Y. Gao and L. Wang, "Sampled-data based consensus of continuous-time multi-agent systems with time-varying topology," *IEEE Transactions on Automatic Control*, vol. 56, no. 5, pp. 1226–1231, 2011.

[14] Y. Liu and Y. Jia, "Robust H∞ consensus control of uncertain multi-agent systems with time delays," *International Journal of Control, Automation and Systems*, vol. 9, no. 6, pp. 1086–1094, 2011.

[15] Y. Zhao, Z. Li, and Z. Duan, "Distributed consensus tracking of multi-agent systems with nonlinear dynamics under a reference leader," *International Journal of Control*, vol. 86, no. 10, pp. 1859–1869, 2013.

[16] Y. Hu, H. Su, and J. Lam, "Adaptive consensus with a virtual leader of multiple agents governed by locally Lipschitz nonlinearity," *International Journal of Robust and Nonlinear Control*, vol. 23, no. 9, pp. 978–990, 2013.

[17] W. Yu, W. Ren, W. X. Zheng, G. Chen, and J. Lü, "Distributed control gains design for consensus in multi-agent systems with second-order nonlinear dynamics," *Automatica*, vol. 49, no. 7, pp. 2107–2115, 2013.

[18] Y. Zhao, B. Li, J. Qin, H. Gao, and H. R. Karimi, "Consensus and synchronization of nonlinear systems based on a novel fuzzy model," *IEEE Transactions on Cybernetics*, vol. 43, no. 6, pp. 2157–2169, 2013.

[19] Y. Pan, M. J. Er, D. Huang, and Q. Wang, "Adaptive fuzzy control with guaranteed convergence of optimal approximation error," *IEEE Transactions on Fuzzy Systems*, vol. 19, no. 5, pp. 807–818, 2011.

[20] Y. P. Pan, H. Y. Yu, and M. J. Er, "Adaptive neural PD control with semiglobal asymptotic stabilization guarantee," *IEEE Transactions on Neural Networks and Learning Systems*, 2014.

[21] Z. Peng, D. Wang, G. Sun, and H. Wang, "Distributed cooperative stabilisation of continuous-time uncertain nonlinear multi-agent systems," *International Journal of Systems Science*, vol. 44, pp. 1–11, 2013.

[22] A. Das and F. L. Lewis, "Distributed adaptive control for synchronization of unknown nonlinear networked systems," *Automatica*, vol. 46, no. 12, pp. 2014–2021, 2010.

[23] A. Das and F. L. Lewis, "Cooperative adaptive control for synchronization of second-order systems with unknown nonlinearities," *International Journal of Robust and Nonlinear Control*, vol. 21, no. 13, pp. 1509–1524, 2011.

[24] H. Zhang and F. L. Lewis, "Adaptive cooperative tracking control of higher-order nonlinear systems with unknown dynamics," *Automatica*, vol. 48, no. 7, pp. 1432–1439, 2012.

[25] Z. Peng, D. Wang, H. Zhang, G. Sun, and H. Wang, "Distributed model reference adaptive control for cooperative tracking of uncertain dynamical multi-agent systems," *IET Control Theory & Applications*, vol. 7, no. 8, pp. 1079–1087, 2013.

[26] Y. Hong, G. Chen, and L. Bushnell, "Distributed observers design for leader-following control of multi-agent networks," *Automatica*, vol. 44, no. 3, pp. 846–850, 2008.

[27] H. Zhang, F. L. Lewis, and A. Das, "Optimal design for synchronization of cooperative systems: state feedback, observer and output feedback," *IEEE Transactions on Automatic Control*, vol. 56, no. 8, pp. 1948–1952, 2011.

[28] Z. Li, X. Liu, P. Lin, and W. Ren, "Consensus of linear multi-agent systems with reduced-order observer-based protocols," *Systems & Control Letters*, vol. 60, no. 7, pp. 510–516, 2011.

[29] K. Hengster-Movric and F. Lewis, "Cooperative observers and regulators for discrete-time multiagent systems," *International Journal of Robust and Nonlinear Control*, vol. 23, no. 14, pp. 1545–1562, 2013.

[30] Z. Meng and Z. Lin, "On distributed finite-time observer design and finite-time coordinated tracking of multiple double integrator systems via local interactions," *International Journal of Robust and Nonlinear Control*, 2013.

[31] B. Zhou, C. Xu, and G. Duan, "Distributed and truncated reduced-order observer based ouput feedback consensus of multi-agent systems," *IEEE Transactions on Automatic Control*, 2014.

Hindawi Publishing Corporation
Mathematical Problems in Engineering
Volume 2014, Article ID 171958, 11 pages
http://dx.doi.org/10.1155/2014/171958

Research Article

A SLAM Algorithm Based on Adaptive Cubature Kalman Filter

Fei Yu,[1] Qian Sun,[1,2] Chongyang Lv,[3] Yueyang Ben,[1] and Yanwei Fu[3]

[1] *College of Automation, Harbin Engineering University, Harbin, Heilongjiang 150001, China*
[2] *Department of Earth and Space Science and Engineering, York University, Toronto, ON, Canada M3J 1P3*
[3] *College of Science, Harbin Engineering University, Harbin, Heilongjiang 150001, China*

Correspondence should be addressed to Qian Sun; qsun@hrbeu.edu.cn and Yueyang Ben; byy@hrbeu.edu.cn

Received 2 February 2014; Revised 3 April 2014; Accepted 8 April 2014; Published 7 May 2014

Academic Editor: Dongbing Gu

We need to predict mathematical model of the system and a priori knowledge of the noise statistics when traditional simultaneous localization and mapping (SLAM) solutions are used. However, in many practical applications, prior statistics of the noise are unknown or time-varying, which will lead to large estimation errors or even cause divergence. In order to solve the above problem, an innovative cubature Kalman filter-based SLAM (CKF-SLAM) algorithm based on an adaptive cubature Kalman filter (ACKF) was established in this paper. The novel algorithm estimates the statistical parameters of the unknown system noise by introducing the Sage-Husa noise statistic estimator. Combining the advantages of the CKF-SLAM and the adaptive estimator, the new ACKF-SLAM algorithm can reduce the state estimated error significantly and improve the navigation accuracy of the SLAM system effectively. The performance of this new algorithm has been examined through numerical simulations in different scenarios. The results have shown that the position error can be effectively reduced with the new adaptive CKF-SLAM algorithm. Compared with other traditional SLAM methods, the accuracy of the nonlinear SLAM system is significantly improved. It verifies that the proposed ACKF-SLAM algorithm is valid and feasible.

1. Introduction

The simultaneous localization and mapping (SLAM) is that the mobile robot builds up the environment map in the unknown environment by utilizing the sensors onboard; at the same time the robots location is computed by the same map [1–3]. The SLAM was first proposed by Smith, Self, and Cheeseman in 1987, estimating robots poses and the features of the environment simultaneously using extend Kalman filter (EKF) [4]. Since then, the SLAM has been implemented in a number of different domains from indoor robots to outdoor, underwater, ground, and airborne systems [5, 6]. The SLAM algorithm has received extensive attention.

SLAM problem involves unknown and uncertain environment description and sensor noise; therefore its essence is a probabilistic estimation issue which is widely accepted by numerous researchers now [4]. At present, the most typical and widely used SLAM algorithm is EKF-based SLAM (EKF-SLAM) introduced in [7–9]. But the precision of EKF-SLAM is limited because the Taylor expansion will causetruncation errors; on the other hand the EKF needs to calculate the fussy Jacobian matrix which increases the computational load. To solve the problems of the EKF, the unscented Kalman filter (UKF) proposed by Julier and Uhlmann [10] was used in SLAM [11]. Unlike the EKF, the UKF uses a set of chosen samples to represent the state distribution. The UKF-based SLAM (UKF-SLAM) algorithm can not only avoid the calculation of the Jacobian and Hessian matrices but also obtain higher approximation accuracy with the unscented transformation (UT). However, for high-dimensional systems, the computation load is still heavy; thus, the filter converges slowly. In 2009, Arasaratnam and Haykin [12, 13] proposed a more accurate nonlinear filtering solution based on a cubature transform named the cubature Kalman filter (CKF) which can avoid linearization of the nonlinear system by using cubature point sets to approximate the mean and variance. The third-order accuracy of the system can be achieved with this method. What is more, the computation complexity of the CKF is lower than the one of the UKF. Because of its high accuracy and low calculation load, the

CKF is widely used in attitude estimation and navigation [14]. CKF-SLAM was introduced in [5], which achieved high positioning accuracy compared with traditional SLAM algorithms.

However, the above filters are all based on the framework of the Kalman filter (KF); it can only achieve a good performance under the assumption that the complete and exact information of the noise distribution has to be known as a prior. But in practice, the prior noise statistic is usually unknown totally or may be always time-varying because the model and the noise of system are influenced by the drift errors of the robots component, the kinematic model of the robot, and the uncertainty of the outdoor environment. With the uncertain or time-varying noise statistic, the state estimation will have large errors, or, even, the filters will be possible to diverge [15]. For SLAM, it has been shown that the performance can become significantly worse and the estimated results practical diverge with an incorrect a priori knowledge of the noise statistics [16, 17]. Therefore, to solve this problem, many researchers have proposed the H_∞ filter algorithm and the adaptive filter algorithm which have been focused on in recent years. Although the H_∞ filter has numerous advantages, such as being robust, independent of the noise statistic, and easy to set the filter parameters, the low filter accuracy is still its fatal flaw [18–20]. Hence, we only focus on the adaptive filter algorithm in this work.

The core idea of the adaptive filter algorithm is that, while the whole system is filtering, estimating and modifying the parameters of the system model and the statistic of the system noise which are unknown or time-varying, to decrease state estimation errors and improve the estimation accuracy. There are numerous adaptive algorithms, including the distribution test method, the maximum a posteriori method, the correlation test method, and the Sage-Husa adaptive method [21, 22]. Therein, the Sage-Husa adaptive method which is suboptimal unbiased maximum a posteriori (MAP) noise estimator is one of the widely used adaptive filtering methods [23], since it has advantages that the recursive formula is simple and the principle is clear and easy to implement. Therefore we use this kind of adaptive filtering to estimate unknown system noise here.

In this paper a novel adaptive filtering algorithm based on the CKF-SLAM, named adaptive CKF-SLAM (ACKF-SLAM), for mobile robot outdoors, was proposed. In this proposed algorithm, we introduced the Sage-Husa adaptive filtering method into the traditional CKF-SLAM algorithm. The system noise and the statistics of the observation noise were estimated on real time and modified to reduce the errors of the system model by using the adaptive estimator of the time-varying noise. What is more, it also prevented the filter divergence and improved the estimation accuracy significantly. The rest of the paper was organized as follows. The description of the SLAM and the traditional CKF-SLAM algorithm were presented in Section 2. Section 3 showed the adaptive EKF algorithm and then a new ACKF-SLAM algorithm was proposed. Numerical examples in different scenarios along with specific analysis were given in Section 4. Section 5 concluded this paper.

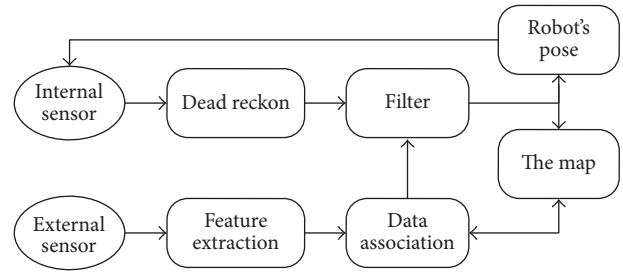


Figure 1: Sketch of the SLAM algorithm.

2. SLAM

2.1. Description of the SLAM. The SLAM is always described as follows: the mobile robot starting in an unknown location without previous knowledge of the environment builds a map using its onboard sensors while, simultaneously, using this same map to determine the location of the robot within this map [2]. Figure 1 illustrates the characteristic data acquisition and processing of the SLAM.

The essence of the SLAM is the filtering estimation of the system in the whole path. The system states are composed of the robots pose (position, orientation) and the landmarks states observed in the environment. Let the robot states be represented by its pose with an estimated vector X_v and the covariance matrix P_{vv}, defined as

$$X_v = \begin{bmatrix} x_v & y_v & \varphi_v \end{bmatrix}^T,$$
$$P_{vv} = \begin{bmatrix} \sigma^2_{x_v x_v} & \sigma^2_{x_v y_v} & \sigma^2_{x_v \varphi_v} \\ \sigma^2_{x_v y_v} & \sigma^2_{y_v y_v} & \sigma^2_{y_v \varphi_v} \\ \sigma^2_{x_v \varphi_v} & \sigma^2_{y_v \varphi_v} & \sigma^2_{\varphi_v \varphi_v} \end{bmatrix}. \tag{1}$$

Without loss of generality, let us assume that there are a set of n 2D static point features observed in the map; the position estimations of these features are given by their estimated vector X_m and the covariance matrix P_{mm} as

$$X_m = \begin{bmatrix} x_1 & y_1 & \cdots & x_n & y_n \end{bmatrix}^T,$$
$$P_{mm} = \begin{bmatrix} \sigma^2_{x_1 x_1} & \sigma^2_{x_1 y_1} & \cdots & \sigma^2_{x_1 x_n} & \sigma^2_{x_1 y_n} \\ \sigma^2_{x_1 y_1} & \sigma^2_{y_1 y_1} & \cdots & \sigma^2_{y_1 x_n} & \sigma^2_{y_1 y_n} \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ \sigma^2_{x_1 x_n} & \sigma^2_{y_1 x_n} & \cdots & \sigma^2_{x_n x_n} & \sigma^2_{x_n y_n} \\ \sigma^2_{x_1 y_n} & \sigma^2_{y_1 y_n} & \cdots & \sigma^2_{x_n y_n} & \sigma^2_{y_n y_n} \end{bmatrix}. \tag{2}$$

In the SLAM, the total state vector X is composed of the robots states X_v and the landmarks states X_m. Hence the

estimations of the total state vector X and the corresponding total error covariance matrix P are given as

$$X = \begin{bmatrix} X_v^T & X_m^T \end{bmatrix}^T, \quad P = \begin{bmatrix} P_{vv} & P_{vm} \\ P_{vm}^T & P_{mm} \end{bmatrix}, \tag{3}$$

wherein P_{vm} denotes the robot-map correlation.

The key of the SLAM is to determine the posterior probability density of the state vector X which is $p(X_{v,k}, X_m \mid u_k, z_k)$ and the implication is to obtain the joint probability density of the robot pose $X_{v,k}$ and the map X_m when the control input u_k and observation z_k are known. We can obtain the following equation by utilizing the Bayes formula:

$$\begin{aligned} & p\left(X_{v,k}, X_m \mid u_k, z_k\right) \\ & \quad = p\left(z_k \mid X_{v,k}, X_m\right) \\ & \quad \times \int p\left(X_{v,k} \mid X_{v,k-1}, u_k\right) \\ & \qquad \times p\left(X_{v,k-1}, X_m \mid u_{k-1}, z_{k-1}\right) dX_{v,k-1}, \end{aligned} \tag{4}$$

wherein u_k is the input control of the robot at time k which drives from pose $X_{v,k-1}$ at time $k-1$ to pose $X_{v,k}$ at time k; z_k is observation to the landmark at time k; $p(X_{v,k} \mid X_{v,k-1}, u_k)$ is the motion model which is the conditional probability of $X_{v,k}$ at time k when u_k and $X_{v,k-1}$ are known; $p(z_k \mid X_{v,k}, X_m)$ is the observation model which is the conditional probability of z_k at time k when $X_{v,k}$ and the landmark collection matrix X_m are known.

2.2. CKF-SLAM. In this subsection the principle of the classical CKF is introduced. Consider the general discrete-time nonlinear state-space model as follows:

$$\begin{aligned} x_k &= f\left(x_{k-1}\right) + W_{k-1}, \\ z_k &= h\left(x_k\right) + \eta_k, \end{aligned} \tag{5}$$

wherein x_k and z_k are the state vector and the measurement vector at time k, respectively; $f(\cdot)$ and $h(\cdot)$ are specific known nonlinear functions; and W_{k-1} and η_k are the noise vectors from two independent Gaussian processes with their means being q_{k-1} and r_k and their covariance matrices being Q_{k-1} and R_k, respectively. The CKF is proposed to solve this nonlinear filtering problem on the basis of the spherical-radial cubature criterion. Firstly it approximates the mean and variance of the probability distribution through a set of $2N$ (N is the dimension of the nonlinear system) cubature points with the same weight, propagates the above cubature points through the nonlinear functions, and then calculates the mean and variance of the current approximate Gaussian distribution by the propagated cubature points [12]. A set of $2N$ cubature points are given by $[\xi_i, \omega_i]$, where ξ_i is the ith cubature point and ω_i is the corresponding weight:

$$\begin{aligned} \xi_i &= \sqrt{N}[1]_i, \\ \omega_i &= \frac{1}{2N}, \end{aligned} \tag{6}$$

wherein $i = 1, 2, \ldots, 2N$. Under the assumption that the posterior density at time $k-1$ is known, the steps involved in the time-update and the measurement-update of the CKF are summarized as follows [12].

Time-update:

(1) factorize

$$P_{k-1|k-1} = S_{k-1|k-1} S_{k-1|k-1}^T, \tag{7}$$

(2) evaluate the cubature points ($i = 1, 2, \ldots, 2N$)

$$X_{i,k-1|k-1} = S_{k-1|k-1}\xi_i + \widehat{x}_{k-1}, \tag{8}$$

(3) evaluate the propagated cubature points

$$X_{i,k|k-1}^* = f\left(X_{i,k-1|k-1}\right) + q_{k-1}, \tag{9}$$

(4) estimate the predicted state

$$\widehat{x}_{k|k-1} = \frac{1}{2N}\sum_{i=1}^{2N} X_{i,k|k-1}^*, \tag{10}$$

(5) estimate the predicted error covariance

$$P_{k|k-1} = \frac{1}{2N}\sum_{i=1}^{2N} X_{i,k|k-1}^* X_{i,k|k-1}^{*T} - \widehat{x}_{k|k-1}\widehat{x}_{k|k-1}^T + Q_{k-1}. \tag{11}$$

Measurement-update:

(1) factorize

$$P_{k|k-1} = S_{k|k-1} S_{k|k-1}^T, \tag{12}$$

(2) evaluate the cubature points

$$X_{i,k|k-1} = S_{k|k-1}\xi_i + \widehat{x}_{k|k-1}, \tag{13}$$

(3) evaluate the propagated cubature points

$$Y_{i,k|k-1} = h\left(X_{i,k|k-1}\right) + r_k, \tag{14}$$

(4) estimate the predicted measurement

$$\widehat{y}_{k|k-1} = \frac{1}{2N}\sum_{i=1}^{2N} Y_{i,k|k-1}, \tag{15}$$

(5) estimate the innovation covariance matrix

$$P_{k|k-1}^{zz} = \frac{1}{2N}\sum_{i=1}^{2N} Y_{i,k|k-1} Y_{i,k|k-1}^T - \widehat{y}_{k|k-1}\widehat{y}_{k|k-1}^T + R_k, \tag{16}$$

(6) estimate the cross-covariance matrix

$$P_{k|k-1}^{xz} = \frac{1}{2N}\sum_{i=1}^{2N} X_{i,k|k-1} Y_{i,k|k-1}^{T} - \hat{x}_{k|k-1}\hat{y}_{k|k-1}^{T}. \quad (17)$$

With the new measurement vector z_k, the estimation of the state vector $\hat{x}_{k|k}$ and its covariance matrix $P_{k|k}$ at time k can be obtained as follows.

(1) Estimate the Kalman gain

$$K_k = P_{k|k-1}^{xz}\left(P_{k|k-1}^{zz}\right)^{-1}. \quad (18)$$

(2) Estimate the updated state

$$\hat{x}_k = \hat{x}_{k|k-1} + K_k\left(z_k - \hat{z}_{k|k-1}\right). \quad (19)$$

(3) Estimate the corresponding error covariance

$$P_{k|k} = P_{k|k-1} - K_k P_{k|k-1}^{zz} K_k^{T}. \quad (20)$$

The CKF-SLAM algorithm uses cubature rule and $2N$ cubature point sets $[\xi_i, \omega_i]$ to compute the mean and variance of the probability distribution without any linearization of the SLAM system. Thus, the CKF-SLAM algorithm does not demand to calculate Jacobian and Hessian matrices so that the truncation errors can be avoided. Hence, the estimation accuracy can reach the third order or higher. Furthermore, the computational complexity is alleviated to a certain extent than the UKF [13].

3. SLAM Algorithm Based on Adaptive CKF

3.1. Adaptive EKF. When the system noises are unknown or time-varying, the filtering algorithm cannot be recursively carried out in a common way. One effective solution is the adaptive filtering algorithm. The adaptive filtering technology has become a focus of the research attempting to solve the filter divergence problem caused by the inaccurate statistical properties of the noise and the mathematical model itself. The Sage-Husa adaptive filtering proposed by Sage and Husa is one of the most widely used adaptive filtering algorithms. Estimating the statistical parameters of the virtual noises online, along with the recursive estimate of the state, this method can reduce the error of the model and improve the accuracy of the whole system validly.

Nowadays, the adaptive EKF based on Sage-Husa is the most widely used adaptive algorithm. However, the estimated accuracy is low because of the truncation errors of the Taylor expansion in the EKF. Considering the advantages of the CKF, an adaptive CKF algorithm was proposed to improve the estimated accuracy in this work on the basis of the adaptive EKF. First of all, the adaptive EKF was introduced in detail here. Suppose that the discrete state equation and the observation equation of the nonlinear system are shown as (5). On the basis of [24], the Sage-Husa estimator based adaptive EKF algorithm is shown as

$$\begin{aligned}
\hat{x}_{k|k-1} &= f\left(\hat{x}_{k-1}\right) + \hat{q}_{k-1},\\
P_{k|k-1} &= F_{k-1}P_{k-1}F_{k-1}^{T} + \widehat{Q}_{k-1},\\
\xi_k &= z_k - h\left(\hat{x}_{k|k-1}\right) - \hat{r}_k,\\
S_k &= H_k P_{k|k-1} H_k^{T} + \widehat{R}_k,\\
K_k &= P_{k|k-1}H_k^{T}S_k^{-1},\\
\hat{x}_k &= \hat{x}_{k|k-1} + K_k\xi_k,\\
P_k &= \left(I - K_k H_k\right)P_{k|k-1},
\end{aligned} \quad (21)$$

wherein P_k is the error covariance matrix of the state $\hat{x}_k$, K_k is the filter gain matrix, F_{k-1} and H_k are the nonlinear states function $f(\cdot)$ and observation function $h(\cdot)$ with respect to states, respectively, ξ_k is the difference between the measurement and the prediction, and S_k covariance matrix of the ξ_k, $\hat{q}_{k-1}$, $\hat{r}_k$, $\widehat{Q}_{k-1}$, and $\widehat{R}_k$ is obtained by recurrence of the time-varying noise statistics estimator:

$$\begin{aligned}
\hat{q}_k &= \left(1 - d_{k-1}\right)\hat{q}_{k-1} + d_{k-1}\left(\hat{x}_k - f\left(\hat{x}_{k-1}\right)\right),\\
\widehat{Q}_k &= \left(1 - d_{k-1}\right)\widehat{Q}_{k-1}\\
&\quad + d_{k-1}\left[K_k\xi_k\xi_k^{T}K_k^{T} + P_k - F_{k-1}P_{k-1}F_{k-1}^{T}\right],\\
\hat{r}_k &= \left(1 - d_{k-1}\right)\hat{r}_{k-1} + d_{k-1}\left(z_k - h\left(\widehat{X}_{k|k-1}\right)\right),\\
\widehat{R}_k &= \left(1 - d_{k-1}\right)\widehat{R}_{k-1} + d_{k-1}\left[\xi_k\xi_k^{T} - H_k P_{k|k-1}H_k^{T}\right],\\
d_{k-1} &= \frac{(1-b)}{\left(1 - b^{k}\right)},
\end{aligned} \quad (22)$$

wherein ξ_{k+1} denotes the innovation sequence; b $(0 < b < 1)$ denotes the forgetting factor whose value is often set between 0.95 and 0.99. The memory span is limited utilizing the forgetting factor. As a result, the old information is forgotten little by little and the new information plays the lead role in estimating.

It has been analyzed that adaptive filtering algorithms cannot estimate the process and the measurement noise simultaneously in [25]. However, theoretical derivations and simulations in [26] show that when the measurement noise is already known, the process noise can be obtained by the iteration. Usually, the measurement noise statistic is relatively well known compared to the system model noise. So we can use this adaptive estimator to estimate the system noise and enhance the filtering accuracy.

3.2. Adaptive CKF. It is well known that the truncation errors of the Taylor expansion in the EKF will reduce the estimated accuracy or even diverge the filter. However, the new algorithm named the CKF can increase the estimated accuracy effectively, which was referred to in Section 2.2. So analogous with the adaptive EKF, the Sage-Husa noise

statistics estimator was embedded into the CKF-SLAM. And the adaptive CKF-SLAM algorithm was obtained, which combines the advantages of the CKF-SLAM and the Sage-Husa adaptive method. Without loss of generality, we still consider the nonlinear system shown as (5) and the detailed algorithm of the ACKF-SLAM is given as follows.

Time-update:

$$\begin{aligned}
P_{k-1|k-1} &= S_{k-1|k-1}S_{k-1|k-1}^{T},\\
X_{i,k-1|k-1} &= S_{k-1|k-1}\xi_i + \widehat{x}_{k-1},\\
X_{i,k|k-1}^{*} &= f\left(X_{i,k-1|k-1}\right) + \widehat{q}_{k-1},\\
\widehat{x}_{k|k-1} &= \frac{1}{2N}\sum_{i=1}^{2N} X_{i,k|k-1}^{*},\\
P_{k|k-1} &= \frac{1}{2N}\sum_{i=1}^{2N} X_{i,k|k-1}^{*}X_{i,k|k-1}^{*T} - \widehat{x}_{k|k-1}\widehat{x}_{k|k-1}^{T} + \widehat{Q}_{k-1}.
\end{aligned} \tag{23}$$

Measurement-update:

$$\begin{aligned}
P_{k|k-1} &= S_{k|k-1}S_{k|k-1}^{T},\\
X_{i,k|k-1} &= S_{k|k-1}\xi_i + \widehat{x}_{k|k-1},\\
Y_{i,k|k-1} &= h\left(X_{i,k|k-1}\right) + \widehat{r}_k,\\
\widehat{z}_{k|k-1} &= \frac{1}{2N}\sum_{i=1}^{2N} Y_{i,k|k-1},\\
P_{k|k-1}^{zz} &= \frac{1}{2N}\sum_{i=1}^{2N} Y_{i,k|k-1}Y_{i,k|k-1}^{T} - \widehat{z}_{k|k-1}\widehat{z}_{k|k-1}^{T} + \widehat{R}_k,\\
P_{k|k-1}^{xz} &= \frac{1}{2N}\sum_{i=1}^{2N} X_{i,k|k-1}Y_{i,k|k-1}^{T} - \widehat{x}_{k|k-1}\widehat{z}_{k|k-1}^{T}.
\end{aligned} \tag{24}$$

With the new measurement vector z_k, the estimation of the state vector $\widehat{x}_{k|k}$ and its covariance matrix $P_{k|k}$ at time k can be obtained by the following equations:

$$\begin{aligned}
K_k &= P_{k|k-1}^{xz}\left(P_{k|k-1}^{zz}\right)^{-1},\\
\widehat{x}_k &= \widehat{x}_{k|k-1} + K_k\xi_k,\\
\xi_k &= z_k - \widehat{z}_{k|k-1},\\
P_{k|k} &= P_{k|k-1} - K_kP_{k|k-1}^{zz}K_k^{T},\\
\widehat{q}_k &= \left(1-d_{k-1}\right)\widehat{q}_{k-1} + d_{k-1}\left(\widehat{X}_k - \widehat{x}_{k|k-1}\right),\\
\widehat{Q}_k &= \left(1-d_{k-1}\right)\widehat{Q}_{k-1} + d_{k-1}\left[K_k\xi_k\xi_k^{T}K_k^{T} + P_k - P_{k|k-1}\right],\\
\widehat{r}_k &= \left(1-d_{k-1}\right)\widehat{r}_{k-1} + d_{k-1}\left(z_k - \widehat{z}_{k|k-1}\right),\\
\widehat{R}_k &= \left(1-d_{k-1}\right)\widehat{R}_{k-1} + d_{k-1}\left[\xi_k\xi_k^{T} - P_{k|k-1}^{zz}\right],\\
d_{k-1} &= \frac{(1-b)}{\left(1-b^{k}\right)}.
\end{aligned} \tag{25}$$

The improved ACKF-SLAM algorithm proposed in this paper can be used in the situation when the noise statistical character of the system is absolutely or approximately unknown to make sure the filter works well, enhancing the stability of the filter.

4. Experiments and Analysis

4.1. Experiment Modeling. In order to verify the effectiveness and feasibility of the new SLAM algorithm proposed in this paper, a large number of simulations are tested with the simulation environment issued by Tim Bailey from University of Sydney. The simulation environment which is a 250 m × 200 m outdoor area includes the movement path and 135 static landmark points. The mobile robot moves along the path from (0, 0) anticlockwise as it is shown in Figure 2.

The motion model of the mobile robot is shown as follows:

$$x_{v,k+1} = \begin{bmatrix} x_{vx,k} + \Delta T v_{k+1}\cos\left(x_{v\varphi,k} + \theta_{k+1}\right)\\ x_{vy,k} + \Delta T v_{k+1}\cos\left(x_{v\varphi,k} + \theta_{k+1}\right)\\ x_{v\varphi,k} + \dfrac{\Delta T v_{k+1}\sin\theta_{k+1}}{\mathrm{WB}} \end{bmatrix} + W_{k+1}, \tag{26}$$

wherein $[x_{vx,k}, x_{vy,k}, x_{v\varphi,k}]^{T}$ denotes the pose of the mobile robot at time k; $x_{v,k+1}$ denotes the pose of the mobile robot at time $k-1$; ΔT denotes the sampling time; v_{k+1} denotes the velocity of the mobile robot at time $k-1$; θ_{k+1} denotes the azimuth angle at time $k-1$; WB denotes the wheel base between the two axes; W_{k+1} denotes the error of the system at time $k-1$ and $W \sim N(0, Q)$. The observation model is shown as follows:

$$z_k = \begin{bmatrix} r_i\\ \varphi_i \end{bmatrix} = \begin{bmatrix} \sqrt{\left(x_i - x_{vx,k}\right)^2 + \left(y_i - x_{vy,k}\right)^2}\\ \arctan\dfrac{y_i - x_{vy,k}}{x_i - x_{vx,k}} - x_{v\varphi,k} \end{bmatrix} + V_k, \tag{27}$$

wherein (x_i, y_i) denotes the detected position corresponding to the ith feature; r_i denotes the distance between the ith detected feature and the mobile robot; φ_i denotes the distance between the ith detected feature and the heading of the mobile robot; V_k denotes the measurement error at time k and $V \sim N(0, R)$.

The experiment parameters are set as follows. The initial states of the mobile robot $x_{v,0} = [0, 0, 0]^{T}$; the sampling interval $\Delta T = 0.025$ s; the velocity $v = 3$ m/s; the velocity error $\sigma_v = 0.25$ m/s; the azimuth error $\sigma_\theta = 2^{\circ}$; the maximum angular rate is 0.2 s; the maximum distance of measurement is 30 m; the distance error $\sigma_r = 0.2$ m; the angular error $\sigma_\varphi = 1^{\circ}$.

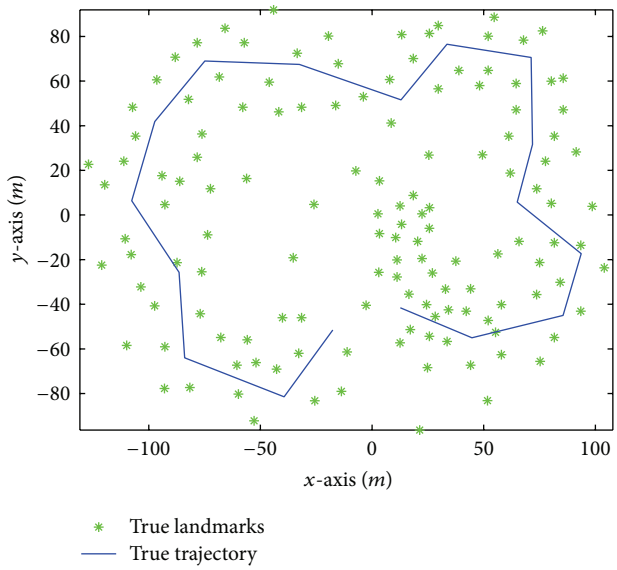


Figure 2: Sketch of the simulation environment.

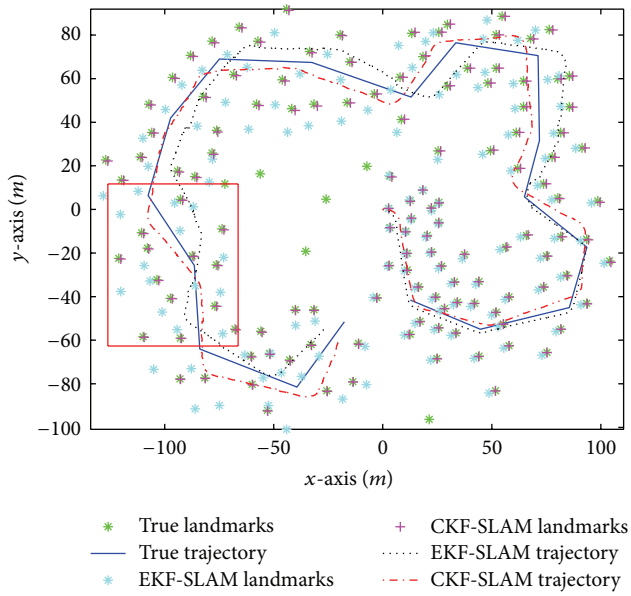


Figure 3: Comparisons of the simulation results with EKF-SLAM and CKF-SLAM.

4.2. Experimental Results and Analysis. Under the above conditions, fifty Monte Carlo simulations were performed for four SLAM algorithms, including the EKF-SLAM, CKF-SLAM, adaptive EKF-SLAM (AEKF-SLAM), and adaptive CKF-SLAM (ACKF-SLAM). And then the root mean square errors (RMSE) of the estimated results were compared.

Figures 3 and 4 show the comparisons of the EKF-SLAM and the CKF-SLAM; Figures 5 and 6 show the comparisons of the AEKF-SLAM and the ACKF-SLAM, wherein Figures 4 and 6 are partial enlargements of the rectangle regions in Figures 3 and 5, respectively. System noise Q and observation noise R are shown as follows:

$$Q = \begin{bmatrix} (0.25)^2 & 0 \\ 0 & \left(\frac{3\pi}{180}\right)^2 \end{bmatrix}, \qquad R = \begin{bmatrix} (0.1)^2 & 0 \\ 0 & \left(\frac{\pi}{180}\right)^2 \end{bmatrix}. \tag{28}$$

From Figures 3 to 6, we obviously know that the estimated error of the ACKF-SLAM is much smaller than that of the CKF-SLAM and EKF-SLAM. It means that the ACKF-SLAM algorithm can provide higher estimated accuracy of the nonlinear SLAM system than the other three solutions.

Figures 7 and 8 show the RMSEs of the estimated position errors of the ACKF-SLAM, AEKF-SLAM, CKF-SLAM, and EKF-SLAM.

From the results shown in Figures 7 and 8, we learn that the RMSE values of the position estimation errors of the ACKF-SLAM algorithm are within 3 m. The precision of AEKF-SLAM is nearly the same as CKF-SLAM, and both of them are lower than ACKF-SLAM algorithm. The RMSE values of the AEKF-SLAM are within 8 m while the corresponding RMSEs of the CKF-SLAM are within 10 m. The EKF-SLAM algorithm has the lowest estimation precision and the RMSEs are within 17 m. So the precision of ACKF-SLAM is higher than the other three algorithms. Comparing between these four algorithms, the estimation

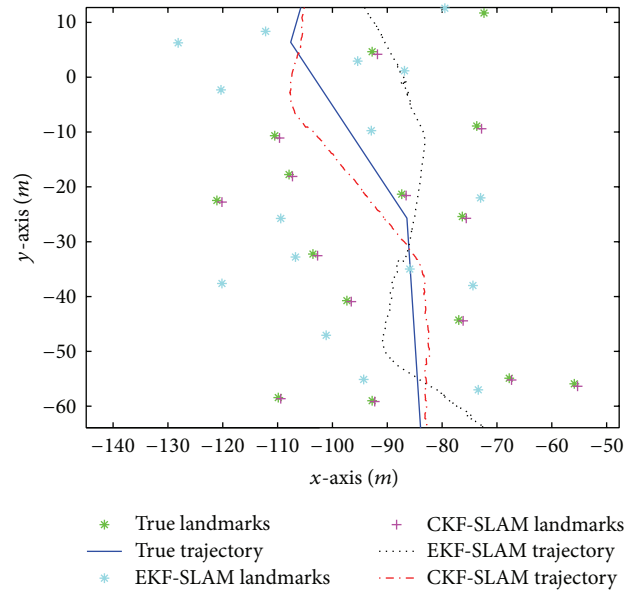


Figure 4: Partial enlargement of Figure 3.

precisions of the standard CKF-SLAM and ACKF-SLAM are higher than those of standard EKF-SLAM and AEKF-SLAM because the cubature points are used in the CKF instead of the nonlinear transformation in the EKF. As a result, the errors can be reduced and the filtering precisions can be enhanced availably. We also can see that the estimation errors increase rapidly in the last part of the standard EKF-SLAM and CKF-SLAM. However, the estimation results of the ACKF-SLAM and AEKF-SLAM always keep stable. It is because the system process noise and observation noise are estimated and modified on real time to enhance the filter

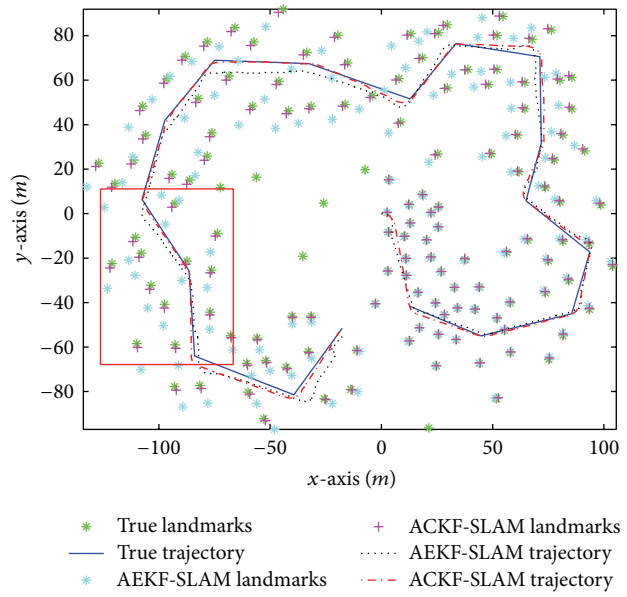


Figure 5: Comparisons of the simulation results with AEKF-SLAM and ACKF-SLAM.

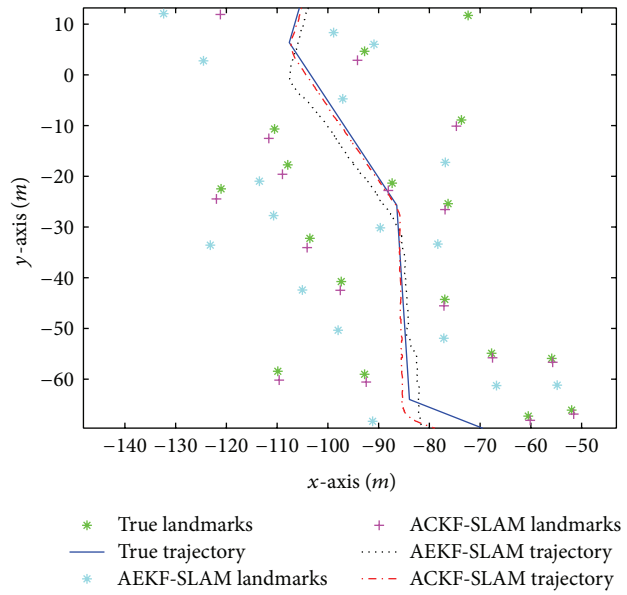


Figure 6: Partial enlargement of Figure 5.

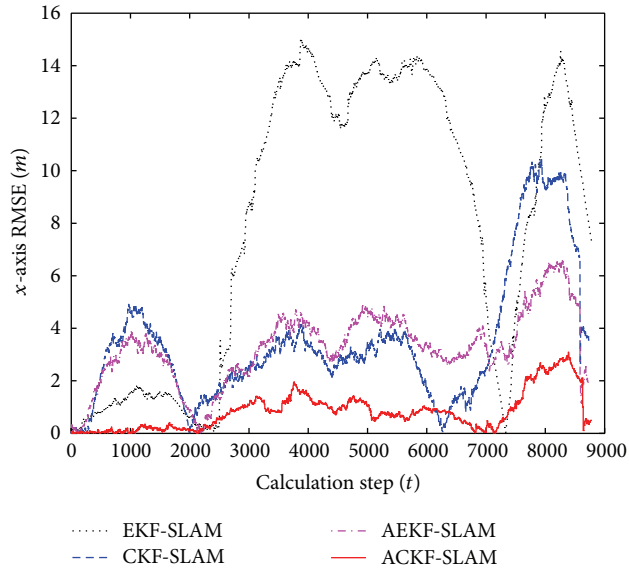


Figure 7: Comparisons of the RMSEs of the x-axis estimated position errors.

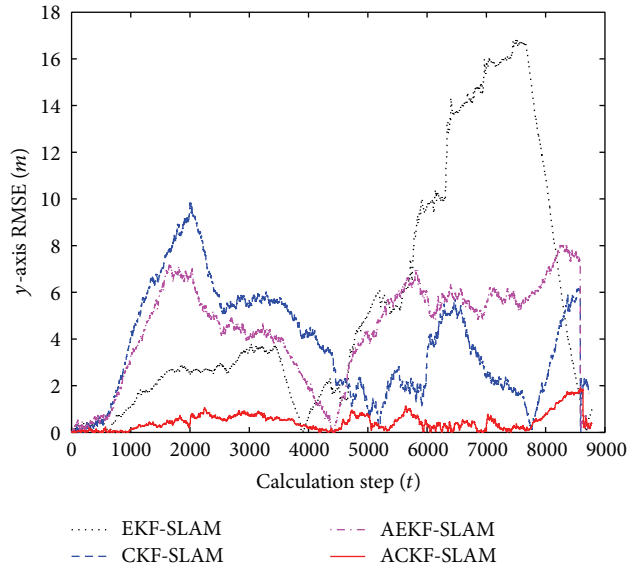


Figure 8: Comparisons of the RMSEs of the y-axis estimated position errors.

precision using the adaptive algorithm. These coincide with the theoretical analysis totally.

Table 1 shows the run time of the four algorithms. The calculated cost of the EKF-SLAM is the least one, the ones of the AEKF-SLAM and the CKF-SLAM are almost equal, and the one of the ACKF-SLAM is the most. That is because of the fact that the EKF only uses the Taylor expansion to linearize the nonlinear system which is extremely simple. But the CKF algorithm approximates the mean and variance by a set of $2N$ cubature points, so it costs a longer time than the EKF. As the adaptive algorithm is adopted in adaptive CKF-SLAM algorithm, it costs much longer time than the CKF-SLAM algorithm.

In order to verify the validity of the ACKF-SLAM sufficiently, some different simulations were done with different environment noise. Suppose that the observa-tion noise obeys the mixture Gauss distribution as $V_k \sim 0.5N(0, R_1) + 0.5N(0, R_2)$, wherein

$$R_1 = \begin{bmatrix} (0.1)^2 & \frac{0.01\pi}{180} \\ \frac{0.01\pi}{180} & \left(\frac{\pi}{180}\right)^2 \end{bmatrix}, \quad R_2 = \begin{bmatrix} (0.01)^2 & \frac{0.3\pi}{180} \\ \frac{0.3\pi}{180} & \left(\frac{3\pi}{180}\right)^2 \end{bmatrix}. \tag{29}$$

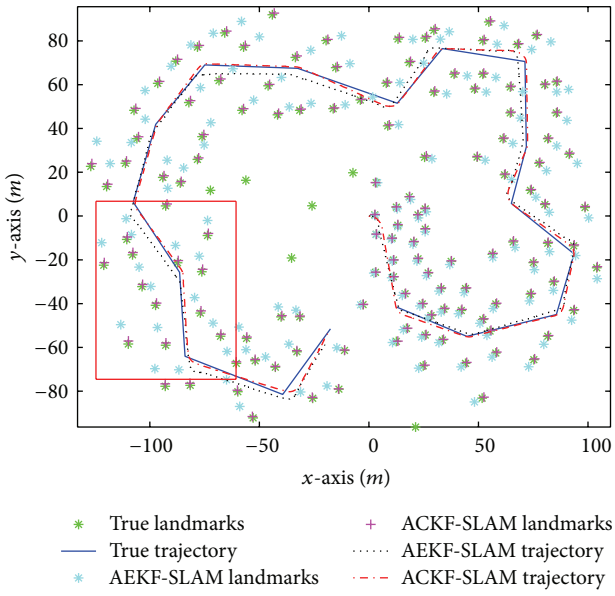


Figure 9: Comparisons of simulation results about the AEKF-SLAM and ACKF-SLAM.

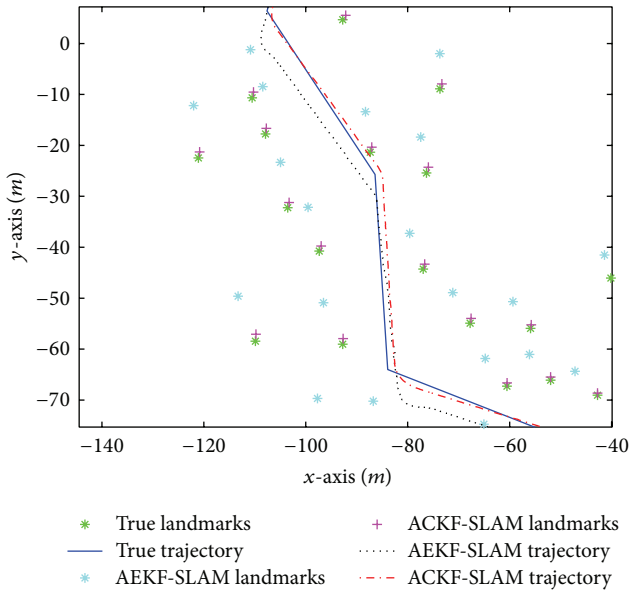


Figure 10: Partial enlargement of Figure 9.

Table 1: Comparisons of the run time.

SLAM algorithm	Run time (seconds)
Standard EKF-SLAM	196.18
Standard CKF-SLAM	278.52
Adaptive EKF-SLAM	231.74
Adaptive CKF-SLAM	344.39

Figure 9 illustrates the estimated errors of the ACKF-SLAM and the AEKF-SLAM algorithms and Figure 10 is the partial enlargement of the rectangle region in Figure 9. As can be seen from Figures 9 to 11, the mobile robot motion trajectory still can be estimated with a high precision by the ACKF-SLAM even though different environment noise is added.

To verify the effectiveness and superiority of the new adaptive SLAM algorithm, when the system noises are time-varying, another simulation scenario was set in which the process and the observation noises are time-varying. On the basis of [2], the parameters are set as Table 2.

Figure 12 illustrates the estimation errors of the ACKF-SLAM, AEKF-SLAM, and CKF-SLAM algorithms when the system noises are time-varying and Figure 13 is the partial enlargement of the rectangle region in Figure 12. Figure 14 is the RMSEs of the position errors correspondingly. As can be seen from Figures 12 to 14, the estimated errors of the CKF-SLAM are the biggest and the ones of the ACKF-SLAM and the AEKF-SLAM are much smaller. That is because of the fact that when the system noises are time-varying, standard CKF-SLAM cannot accurately estimate them. However, the adaptive CKF-SLAM and adaptive EKF-SLAM can estimate and modify the parameters of the system model and the statistic of the system noises online to decrease the estimated errors and improve the accuracy effectively, owing to the Sage-Husa adaptive estimator. Additionally, comparing between the two adaptive methods, the estimated results of the ACKF-SLAM are much better than that of the AEKF-SLAM.

From the simulation results and the theoretical analysis above, we can see that the ACKF-SLAM has higher estimation accuracy with a better numerical performance in the entire motion process than other algorithms. The effectiveness and superiority of the ACKF-SLAM are verified.

5. Conclusions and Future Work

In this paper, a novel nonlinear SLAM algorithm based on the CKF and the Sage-Husa adaptive estimator was proposed in order to solve the problem that the noise statistic of the system is unknown or time-varying in real world and to increase the accuracy of the estimation. Firstly, the Sage-Husa adaptive noise estimator was introduced and the superiority of the CKF solution was analyzed theoretically. Secondly, we focused on proposing a new adaptive SLAM based on the CKF method through combining the advantages of the Sage-Husa adaptive estimator and the CKF-SLAM solution, and the novel ACKF-SLAM algorithm proposed here can estimate and correct the statistical character of the noises in real time and decrease the estimated errors. To verify the new SLAM algorithm, numerical simulations in different scenarios were carried out. The results showed that the proposed adaptive SLAM algorithm based on CKF can not only be applied to the systems in which the noise distribution is unknown or time-varying, but also significantly improve the estimation accuracy of the nonlinear

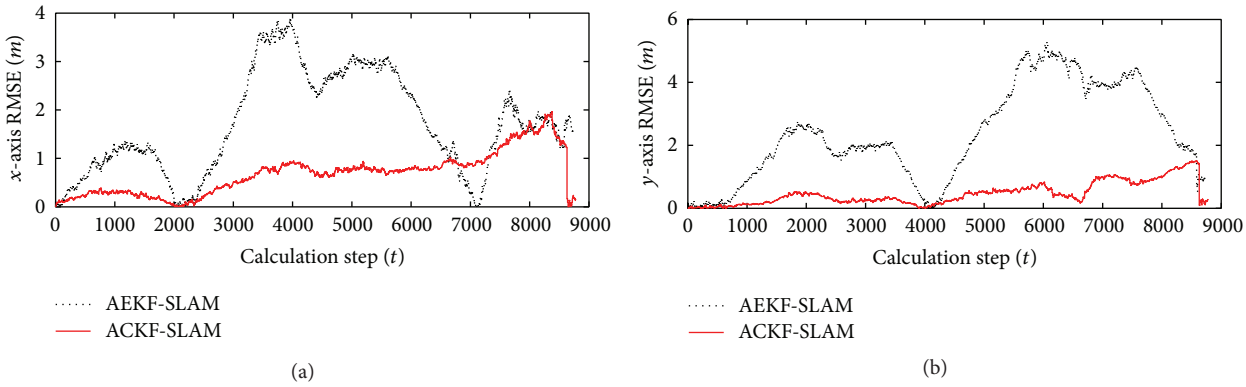


Figure 11: RMSEs of position errors with the AEKF-SLAM and ACKF-SLAM algorithms.

Table 2: Simulation parameters.

Calculation step	Q_k	R_k
$1 < t < 3000$	$\text{diag}\left(\left[(0.1)^2; (0.5\pi/180)^2\right]\right)$	$\text{diag}\left(\left[(0.5)^2; (0.2\pi/180)^2\right]\right)$
$3000 \le t < 6000$	$\text{diag}\left(\left[(0.5)^2; (2\pi/180)^2\right]\right)$	$\text{diag}\left(\left[(0.2)^2; (\pi/180)^2\right]\right)$
$6000 \le t$	$\text{diag}\left(\left[(2)^2; (4\pi/180)^2\right]\right)$	$\text{diag}\left(\left[1^2; (2\pi/180)^2\right]\right)$

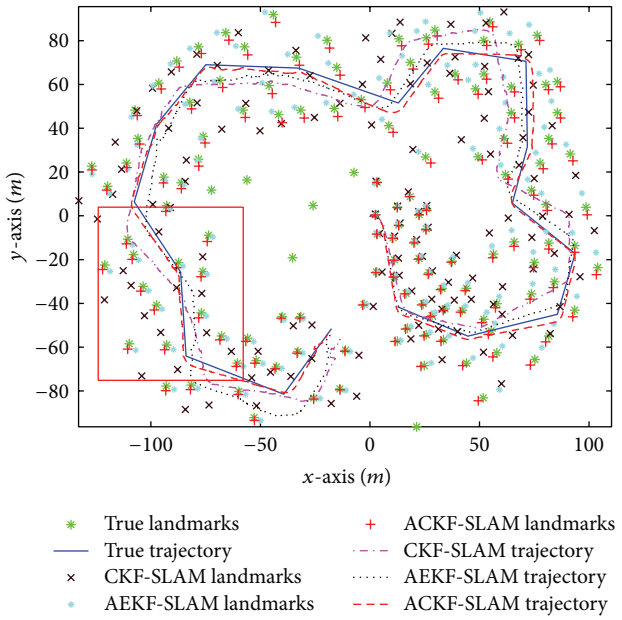


Figure 12: Comparisons of simulation results with time-varying noises.

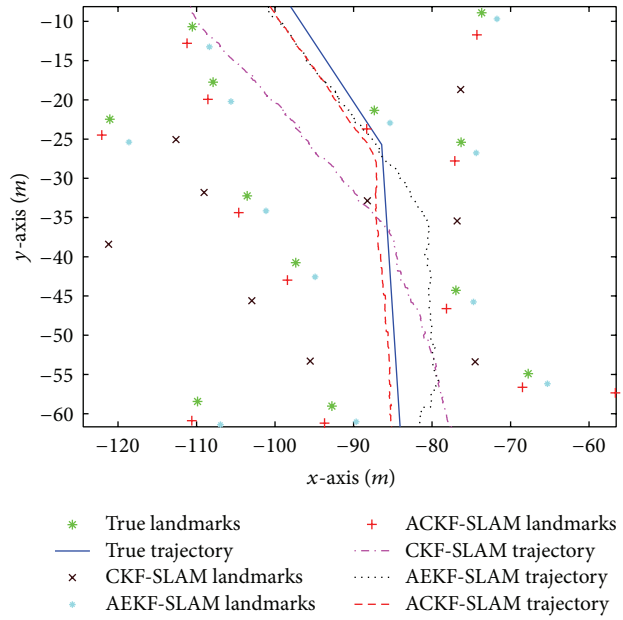


Figure 13: Partial enlargement of Figure 12.

SLAM system than traditional SLAM algorithms. The ACKF-SLAM algorithm provides a new method for simultaneous localization and mapping in an unknown environment. However, the calculation burden of the ACKF algorithm is still huge. Reference [27] presented graph-based SLAM algorithms which can reduce the calculation cost significantly using a graph whose nodes correspond to the poses of the robot at different points in time and whose edges represent constraints between the poses. As a result, we plan to work on the adaptive graph-based SLAM algorithm to reduce the computation time at no sacrifice of the accuracy in the future.

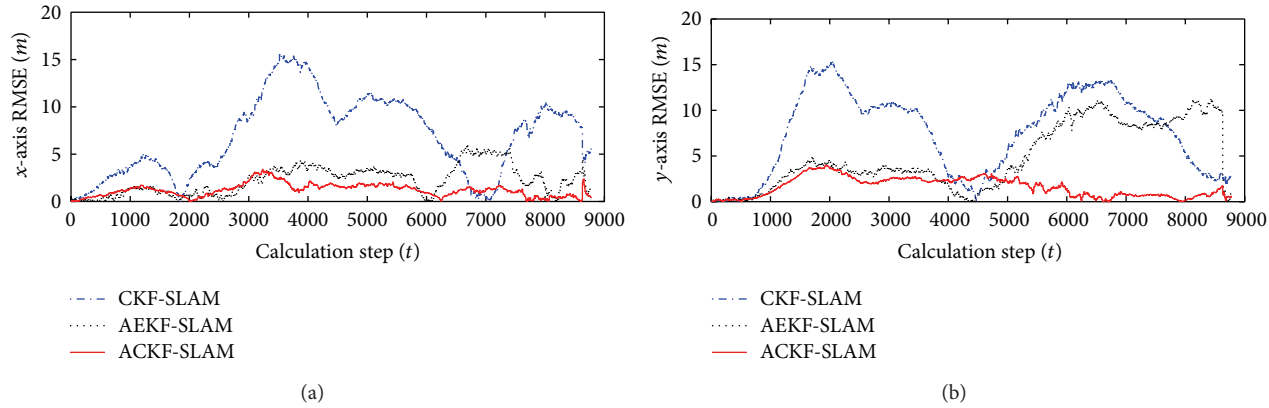


Figure 14: RMSEs of position errors of the CKF-SLAM, AEKF-SLAM, and ACKF-SLAM algorithms.

Conflict of Interests

The authors declare that there is no conflict of interests regarding the publication of this paper.

Acknowledgments

This work was supported by the Fundamental Research Funds for the Central Universities (HEUCFL1411002) and the National Natural Science Foundation of China (51379047).

References

[1] W. Zhang, M. Zhu, and Z. Chen, "An adaptive SLAM algorithm based on strong tracking UKF," *Robot*, vol. 32, no. 2, pp. 190–195, 2010.

[2] H. Wang, G. Fu, J. Li, Z. Yan, and X. Bian, "An adaptive UKF based SLAM method for unmanned underwater vehicle," *Mathematical Problems in Engineering*, vol. 2013, Article ID 605981, 12 pages, 2013.

[3] E. Guerra, R. Munguia, Y. Bolea, and A. Grau, "Validation of data association for monocular slam," *Mathematical Problems in Engineering*, vol. 2013, Article ID 671376, 11 pages, 2013.

[4] H. Durrant-Whyte and T. Bailey, "Simultaneous localization and mapping: part I," *IEEE Robotics and Automation Magazine*, vol. 13, no. 2, pp. 99–108, 2006.

[5] K. P. B. Chandra, D. W. Gu, and I. Postlethwaite, "Cubature kalman filter based localization and mapping," *World Congress*, vol. 18, no. 1, pp. 2121–2125, 2011.

[6] A. Nüchter, H. Surmann, K. Lingemann, J. Hertzberg, and S. Thrun, "6D SLAM with an application in autonomous mine mapping," in *Proceedingsof the IEEE International Conference on Robotics and Automation (ICRA '04)*, vol. 2, pp. 1998–2003, May 2004.

[7] A. Chatterjee and F. Matsuno, "A Geese PSO tuned fuzzy supervisor for EKF based solutions of simultaneous localization and mapping (SLAM) problems in mobile robots," *Expert Systems with Applications*, vol. 37, no. 8, pp. 5542–5548, 2010.

[8] S. Huang and G. Dissanayake, "Convergence and consistency analysis for extended Kalman filter based SLAM," *IEEE Transactions on Robotics*, vol. 23, no. 5, pp. 1036–1049, 2007.

[9] G. Sun, M. Wang, and L. Wu, "Unexpected results of Extended Fractional Kalman Filter for parameter identification in fractional order chaotic systems," *International Journal of Innovative Computing, Information and Control*, vol. 7, no. 9, pp. 5341–5352, 2011.

[10] S. J. Julier and J. K. Uhlmann, "A new extension of the kalman filter to nonlinear systems," in *Proceedings of the International Symposium on Aerospace/Defense, Sensing, Simulation, and Controls*, vol. 3, no. 26, p. 3.2, Orlando, Fla, USA, 1997.

[11] X. Yan, C. Zhao, and J. Xiao, "A novel fastslam algorithm based on iterated unscented kalman filter," in *Proceedings of the IEEE International Conference on Robotics and Biomimetics (ROBIO '11)*, pp. 1906–1911, 2011.

[12] I. Arasaratnam and S. Haykin, "Cubature kalman filters," *IEEE Transactions on Automatic Control*, vol. 54, no. 6, pp. 1254–1269, 2009.

[13] I. Arasaratnam, S. Haykin, and T. R. Hurd, "Cubature Kalman filtering for continuous-discrete systems: theory and simulations," *IEEE Transactions on Signal Processing*, vol. 58, no. 10, pp. 4977–4993, 2010.

[14] W. Gao, Y. Zhang, and J. Wang, "A strapdown interial navigation system/beidou/doppler velocity log integrated navigation algorithm based on a cubature kalman filter," *Sensors*, vol. 14, no. 1, pp. 1511–1527, 2014.

[15] X. Tang, J. Wei, and K. Chen, "Square-root adaptive cubature kalman filter with application to spacecraft attitude estimation," in *Proceedings of the 15th International Conference on Information Fusion*, pp. 1406–1412, 2012.

[16] R. K. Mehra, "On the identification of variances and adaptive Kalman filtering," *IEEE Transactions on Automatic Control*, vol. 15, no. 2, pp. 175–184, 1970.

[17] F. R. Fitzgerald, "Divergence of the Kalman filter," *IEEE Transactions on Automatic Control*, vol. 16, no. 6, pp. 736–747, 1971.

[18] P. Batista, C. Silvestre, and P. J. Oliveira, "Kalman and H infinity optimal filtering for a class of kinematic systems," in *Proceedings of the 17th World Congress, International Federation of Automatic Control (IFAC '08)*, July 2008.

[19] Z. Wang, B. Shen, and X. Liu, "H_∞ filtering with randomly occurring sensor saturations and missing measurements," *Automatica*, vol. 48, no. 3, pp. 556–562, 2012.

[20] J.-H. Wang, C.-L. Song, X.-T. Yao, and J.-B. Chen, "Sigma point H-infinity filter for initial alignment in marine strapdown inertial navigation system," in *Proceedings of the 2nd International Conference on Signal Processing Systems (ICSPS '10)*, vol. 1, pp. V1580–V1584, July 2010.

[21] L. Zhao and X. Wang, "An adaptive UKF with noise statistic estimator," in *Proceedings of the 4th IEEE Conference on Industrial Electronics and Applications (ICIEA '09)*, pp. 614–618, May 2009.

[22] G. Sun, M. Wang, L. Huang, and L. Shen, "Generating multi-scroll chaotic attractors via switched fractional systems," *Circuits, Systems, and Signal Processing*, vol. 30, no. 6, pp. 1183–1195, 2011.

[23] J. Wang, J. Liu, and B.-G. Cai, "Study on information fusion algorithm in embedded integrated navigation system," in *Proceedings of the International Conference on Intelligent Computation Technology and Automation (ICICTA '08)*, vol. 2, pp. 1007–1010, October 2008.

[24] H. Wang, J. Wang, X. Bian, and G. Fu, "SLAM of AUV based on the combined EKF," *Jiqiren/Robot*, vol. 34, no. 1, pp. 56–64, 2012.

[25] S. Kawamura and N. Sakagami, "Planning and control of robot motion based on time-scale transformation," in *Advances in Robot Control*, pp. 157–178, Springer, New York, NY, USA, 2006.

[26] S. Mei, F. Liu, and A. Xue, *The Semi Tensor Product Methods in the Power System Transient Analysis*, Tsinghua University Press, 2010.

[27] G. Grisetti, R. Kummerle, C. Stachniss, and W. Burgard, "A tutorial on graph-based SLAM," *IEEE Intelligent Transportation Systems Magazine*, vol. 2, no. 4, pp. 31–43, 2010.

Hindawi Publishing Corporation
Mathematical Problems in Engineering
Volume 2014, Article ID 835301, 11 pages
http://dx.doi.org/10.1155/2014/835301

Research Article

Multimodel Predictive Control Approach for UAV Formation Flight

Chang-jian Ru,[1] Rui-xuan Wei,[1] Ying-ying Wang,[1] and Jun Che[2]

[1] *Air Force Engineering University, Xi'an 710038, China*
[2] *Science and Technology on Aircraft Control Laboratory, FACRI, Xi'an 710065, China*

Correspondence should be addressed to Chang-jian Ru; ruchangjian1986@gmail.com

Received 20 December 2013; Revised 10 March 2014; Accepted 25 March 2014; Published 6 May 2014

Academic Editor: Leo Chen

Formation flight problem is the most important and interesting problem of multiple UAVs (unmanned aerial vehicles) cooperative control. In this paper, a novel approach for UAV formation flight based on multimodel predictive control is designed. Firstly, the state equation of relative motion is obtained and then discretized. By the geometrical method, the characteristic points of state are determined. Afterwards, based on the linearization technique, the standard linear discrete model is obtained at each characteristic state point. Then, weighted model set is proposed using the idea of T-S (Takagi-Sugeno) fuzzy control and the predictive control is carried out based on the multimodel method. Finally, to verify the performance of the proposed method, two different simulation scenarios are performed.

1. Introduction

In recent years, as an advanced system with high autonomy, UAVs have been widely applied in the fields of both civilian and military. When a single UAV accomplishes tasks individually, it will be more likely to reduce mission success, due to its limited information accessing ability. In comparison, multiple UAVs collaborating with each other maintain a certain formation during the flight, which provides them with full access to environmental information, increases resistance to external attack capability, improves working efficiency and robustness of the system, and so forth, so it has attracted wide attention [1–3]. Formation flight is an important aspect of multiple UAVs cooperative control. When the formation shape maintains or changes according to the mission requirements, it is necessary to control the relative position, attitude, and speed between UAVs, and so forth. However, UAV's control system is a nonlinear coupling system, coupled with complex operational environment constraints, putting forward higher design requirement for formation controller. So it is essential to propose an effective control strategy to solve those problems.

Model predictive control (MPC) method is put forward by some scholars. Here, several typical researches are presented. A hierarchical approach and a set of MPC strategies for the UAV formation are proposed in [4], where obstacle and collision avoidance constraints are taken into account. A distributed collision-free formation flight control law in the framework of nonlinear model predictive control is designed in [5]. In [6], a dual mode MPC method is used for formation control. To guarantee the stability, the dual mode controller must switch from an MPC control to a terminal state controller. A simple nonlinear model predictive control (NMPC) formulation is used to adequately address the terrain avoidance problem, as presented in [7]. An online nonlinear model predictive control framework is used for the trajectory tracking of autonomous vehicles in [8], where a bicycle model is used for the prediction of future states in the NMPC framework. The validation of a formation flight control technique with obstacle avoidance capability based on nonlinear model predictive algorithms is proposed in [9].

The nonlinear model predictive control method provides an effective means to solve the control problem of nonlinear systems [10, 11]. Because the close relative distance between

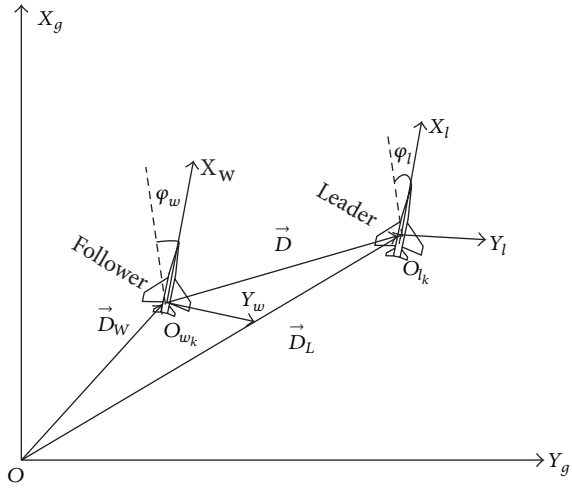


Figure 1: The position relationship between two vehicles.

the UAVs may lead to collision; thus, it requires higher control accuracy. However, stair-like MPC uses the way of constraining the variation of the future control quantity, which restricts the maneuverability of the vehicle and is prone to causing collision between the UAVs due to the overshoot problem [12]. So it is necessary to adopt a new predictive control method to achieve the formation flight control. For some more complex systems, the multimodel control method has stronger robustness and higher control accuracy under certain conditions [13]. Besides, multimodel control method can provide the nonlinear system with transparent model and controller, facilitating the system analysis. Compared with other nonlinear global strategies, the multimodel control method cannot greatly reduce computational complexity, but the model and structure of controller are more suitable for online adjustments and learning algorithm [14], so multiple model-based predictive control can be used to solve UAV formation control problem.

This paper is organized as follows. In Section 2, the discrete relative motion equations for UAV formation are established. In Section 3, a multiple models-based predictive control approach is used to design controller of the formation. Simulation results are given in Section 4. Finally, Section 5 concludes the paper.

2. UAV Formation Flight Control Model

2.1. Kinematics Model of UAV Formation Flight Control. Assume that during the formation flight, an UAV is flying horizontally and has no sideslip. In the geographic coordinate system, the relationship between the position vectors of leader UAV (leader) and follower UAV (follower) is shown in Figure 1.

From Figure 1, it is easy to obtain the following equation:

$$\vec{D}_L = \vec{D}_W + \vec{D}, \tag{1}$$

where, $\vec{D}_L$, $\vec{D}_W$ are displacement vectors of two vehicles and $\vec{D}$ is the relative displacement vectors between two vehicles.

Differentiating (1), one can obtain

$$\frac{d\vec{D}_L}{dt} = \frac{d\vec{D}_W}{dt} + \frac{d\vec{D}}{dt}. \tag{2}$$

According to the relationship between the moving coordinate system, one can easily obtain

$$\frac{d\vec{D}_L}{dt} = \frac{\delta\vec{D}_W}{\delta t} + \omega_w \times \vec{D}, \tag{3}$$

where ω_w is the yaw angular rate.

Since the vehicle is supposed to fly horizontally, the equation of motion will be

$$\begin{aligned} \dot{x}_i &= v_i \cos(\varphi_i), \\ \dot{y}_i &= v_i \sin(\varphi_i), \\ \dot{\varphi}_i &= \omega_i, \quad i = l, w, \end{aligned} \tag{4}$$

where the subscript l and w denote leader UAV and follower UAV, respectively.

Combing (3) and (4), the relative motion equation of two vehicles can be obtained as

$$v_l \begin{bmatrix} \cos\varphi_l \\ \sin\varphi_l \end{bmatrix} = v_w \begin{bmatrix} \cos\varphi_w \\ \sin\varphi_w \end{bmatrix} + C_w^L \left(\begin{bmatrix} \dot{x}_d \\ \dot{y}_d \end{bmatrix} + \begin{bmatrix} -\omega_w \cdot y_d \\ \omega_w \cdot x_d \end{bmatrix} \right), \tag{5}$$

where x_d and y_d are X-axis value and Y-axis value of the distance between two vehicles in the tack coordinates of leader UAV, respectively, and C_w^L is the coordinate transformation matrix.

Then, carry on the translational process, and one can obtain

$$\begin{bmatrix} \dot{x}_d \\ \dot{y}_d \end{bmatrix} = C_L^W \left(v_l \begin{bmatrix} \cos\varphi_l \\ \sin\varphi_l \end{bmatrix} - v_w \begin{bmatrix} \cos\varphi_w \\ \sin\varphi_w \end{bmatrix} \right) - \begin{bmatrix} -\omega_w \cdot y_d \\ \omega_w \cdot x_d \end{bmatrix}, \tag{6}$$

where

$$\begin{aligned} C_L^W v_l \begin{bmatrix} \cos\varphi_l \\ \sin\varphi_l \end{bmatrix} &= v_l \begin{bmatrix} \cos\varphi_e \\ \sin\varphi_e \end{bmatrix}, \\ C_L^W v_w \begin{bmatrix} \cos\varphi_w \\ \sin\varphi_w \end{bmatrix} &= \begin{bmatrix} v_w \\ 0 \end{bmatrix}. \end{aligned} \tag{7}$$

So (6) can be written as follows:

$$\begin{bmatrix} \dot{x}_d \\ \dot{y}_d \end{bmatrix} = v_l \begin{bmatrix} \cos(\varphi_l - \varphi_w) \\ \sin(\varphi_l - \varphi_w) \end{bmatrix} - \begin{bmatrix} v_w \\ 0 \end{bmatrix} - \begin{bmatrix} -\omega_w \cdot y_d \\ \omega_w \cdot x_d \end{bmatrix}. \tag{8}$$

Additionally, there is

$$\dot{\varphi}_w = \omega_w. \tag{9}$$

Combing (8) and (9), we can obtain

$$\begin{aligned} \dot{x}_d &= v_l \cos(\varphi_l - \varphi_w) - v_w + \omega_w \cdot y_d, \\ \dot{y}_d &= v_l \sin(\varphi_l - \varphi_w) - \omega_w \cdot x_d, \\ \dot{\varphi}_w &= \omega_w. \end{aligned} \tag{10}$$

Then, (10) can also be written as the state equations which is shown as follows

$$\begin{bmatrix} \dot{x}_d \\ \dot{y}_d \\ \dot{\varphi}_w \end{bmatrix} = \begin{bmatrix} -1 & y_d \\ 0 & -x_d \\ 0 & 1 \end{bmatrix} \begin{bmatrix} v_w \\ \omega_w \end{bmatrix} + \begin{bmatrix} v_l \cos(\varphi_l - \varphi_w) \\ v_l \sin(\varphi_l - \varphi_w) \\ 0 \end{bmatrix}. \quad (11)$$

The output equation is

$$\begin{aligned} y_1 &= x_d, \\ y_2 &= y_d. \end{aligned} \quad (12)$$

2.2. Discrete Model of UAV Formation Movement and Its Predictive Control Analysis. In the previous section, the state equation of formation control is obtained. Here, discretize this equation and the following equation can be obtained

$$\begin{aligned} \begin{bmatrix} x_d(k+1) \\ y_d(k+1) \\ \varphi_w(k+1) \end{bmatrix} = \begin{bmatrix} x_d(k) \\ y_d(k) \\ \varphi_w(k) \end{bmatrix} + \begin{bmatrix} -1 & y_d(k) \\ 0 & -x_d(k) \\ 0 & 1 \end{bmatrix} \begin{bmatrix} v_w(k) \\ \omega_w(k) \end{bmatrix} \Delta T \\ + \begin{bmatrix} v_l(k) \cos(\varphi_l(k) - \varphi_w(k)) \\ v_l(k) \sin(\varphi_l(k) - \varphi_w(k)) \\ 0 \end{bmatrix} \Delta T. \end{aligned} \quad (13)$$

Generally, the sampling periodic time is short during receding optimization process, so the velocity and yaw angle of leader UAV can be considered constant in sampling period [15], which means in a short sampling period, there is

$$\begin{aligned} v_l(k+i) &= v_l(k+i-1) = v_l(k), \\ \varphi_l(k+i) &= \varphi_l(k+i-1) = \varphi_l(k). \end{aligned} \quad (14)$$

According to these two equations, the predicted value of outputs will be obtained as follows:

$$\begin{aligned} &x_{dp}(k+1), x_{dp}(k+2), \ldots, x_{dp}(k+N), \\ &y_{dp}(k+1), y_{dp}(k+2), \ldots, y_{dp}(k+N). \end{aligned} \quad (15)$$

Since at time k, $y_d(k)$, $v_l(k)$, $\varphi_w(k)$, $\varphi_l(k)$ are known, the future output values of formation flight control are $x_{dp}(k+1), x_{dp}(k+2), \ldots, x_{dp}(k+N-1), x_{dp}(k+N)$, and $y_{dp}(k+1), y_{dp}(k+2), \ldots, y_{dp}(k+N-1), y_{dp}(k+N)$, and these outputs are only the function of the future control quantities $v_w(k), v_w(k+1), \ldots, v_w(k+N-1)$ and $\omega_w(k), \omega_w(k+1), \ldots, \omega_w(k+N-1)$. Obviously, this function is a multi-input-multioutput nonlinear control problem. So, these values can be obtained using the receding optimization algorithm. Then, using $v_w(k)$ and $\omega_w(k)$ as outputs of control quantity, and carry out the receding optimization algorithm in sequence, the input of control quantity in the next time can be obtained.

According to Section 1, it can be known that for the problem of UAV formation flight, nonlinear model predictive control and fuzzy stair-like predictive control have some limitations, but the multimodel control method has stronger robustness and higher control accuracy, the final predictive model of which is linear, so the receding optimization problem can be changed from general nonlinear optimization problem to a linear quadratic optimization problem. Since the linear quadratic optimization has faster computation than the ordinary nonlinear optimization, the multiple models can greatly improve real-time of receding optimization [16]. Therefore, multiple model-based predictive control approach is used to design the controller of UAV formation flight.

3. Predictive Control for UAV Formation Based on Multimodel Approach

The basic principle of the multimodel control method is that nearby the different characteristics of nonlinear systems, different linear model is used to describe this nonlinear system, and each linear model only describes a part of nonlinear system dynamics. The multiple linearization models are used to approximate the nonlinear system in its entire operating range, and the controller is designed based on each linearization model. These controllers are combined together to constitute a multimodel controller in some way. Finally, the control of entire nonlinear system can be achieved through coordinated control between multiple linearization models. The basic steps of this method can be summarized in four steps: (1) acquisition of the model set; (2) local linearization of the model set; (3) establishment of the controller set; (4) combination of the model set.

Similarly, the controller design of UAV formation flight can also include these steps, the flow chart of which is shown in Figure 2. The details of controller design will be presented in the following parts using the multimodel prediction method.

3.1. Determination of the State Characteristic Points of Formation Control Model. According to the basic principles of multimodel predictive control, the characteristic state points of the nonlinear model must be obtained first before obtaining multiple models. Characteristic state points and their region are determined using methods in [15, 17]. As is shown in Figure 3, the horizontal axis and vertical axis denote time and trajectory, respectively.

The basic idea is as follows.

Determine the first characteristic state points, then compute the error between reference trajectory tangent through the characteristic point and reference trajectory and compare the error with the maximum permissible error. If it is greater than the maximum permissible error, redetermine the characteristic point to get the next characteristic state point, and calculate repeatedly until the last characteristic state point is obtained; thus, the characteristic state points of nonlinear system and their applicable region can be determined. So the linearization model set can be obtained through the linearization process at the characteristic state point for each region.

For UAV formation control, the characteristic state points are determined as follows.

Assuming that the initial distance between two vehicles are $x_d(0)$ and $y_d(0)$, respectively, when UAV formation control is carried out, so the desired distances of UAV formation

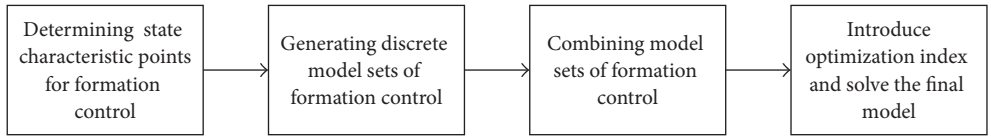


Figure 2: The flow chart of UAV formation flight controller design.

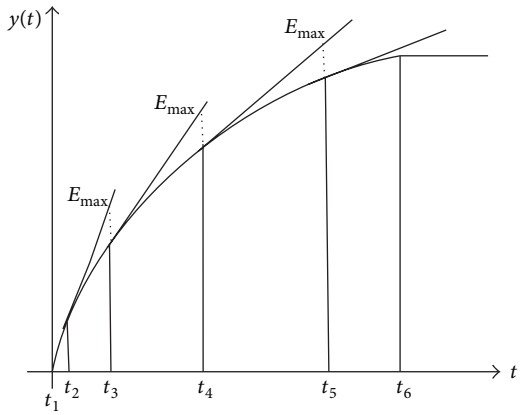


Figure 3: Description of tangent error.

are x_c and y_c. Assuming that the expected time arrival at the desired value of X-axis and Y-axis is the same, the reference trajectory of UAV formation can be obtained as follows:

$$\begin{aligned} y_{1r}(t) &= \xi^t x_d(0) + \left(1-\xi^t\right)\cdot x_c, \\ y_{2r}(t) &= \xi^t y_d(0) + \left(1-\xi^t\right)\cdot y_c. \end{aligned} \tag{16}$$

Since there are two outputs, generating two reference trajectories, the algorithm above cannot be applied directly. But, because the expected arrival time to the desired value is the same, the reference trajectory of an output can be used to determine the characteristic point of state. Assume that the reference trajectory of relative position on the X-axis in the track coordinate system is used to determine characteristic point of the state.

At the characteristic state points, there is

$$\dot{x}_d = 0, \qquad \dot{y}_d = 0, \qquad \dot{\varphi}_w = 0. \tag{17}$$

Assume that ith characteristic state point is $(V_w^i, \omega_w^i, x_d^i, y_d^i, \varphi_d^i)$, then there will be the following equation:

$$\begin{bmatrix} -1 & y_d^i \\ 0 & -x_d^i \\ 0 & 1 \end{bmatrix} \begin{bmatrix} V_w^i \\ \omega_w^i \end{bmatrix} + \begin{bmatrix} V_l \cos\left(\varphi_l - \varphi_w^i\right) \\ V_l \sin\left(\varphi_l - \varphi_w^i\right) \\ 0 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix}. \tag{18}$$

Meanwhile, there is

$$x_d^i = y_{1r}(i), \qquad y_d^i = y_{2r}(i). \tag{19}$$

Solve (18) and (19), then we can obtain

$$\begin{aligned} &\omega_w^i = 0, \qquad \varphi_w^i = \varphi_l, \qquad v_w^i = v_l, \\ &x_d^i = y_{1r}(i), \qquad y_d^i = y_{2r}(i). \end{aligned} \tag{20}$$

Assuming the initial point is as the first characteristic state point, then the state point value will be

$$\begin{aligned} &\omega_w^1 = 0, \qquad \varphi_w^1 = \varphi_l, \qquad v_w^1 = v_l, \\ &x_d^1 = y_{1r}(0), \qquad y_d^1 = y_{2r}(0). \end{aligned} \tag{21}$$

Then, the equation of the tangent is

$$y_{1k}(t) = \dot{y}_{1r}(t)\, t + x_d(0). \tag{22}$$

Afterward, determine the maximum permissible error $E_{\max}$, and the time t_2 corresponding to the second characteristic point can be obtained using the method above. Then, the state point can be obtained as follows

$$\begin{aligned} &\omega_w^2 = 0, \qquad \varphi_w^2 = \varphi_l, \qquad v_w^2 = v_l, \\ &x_d^2 = x_d(t_2), \qquad y_d^2 = y_d(t_2). \end{aligned} \tag{23}$$

So the tangent equation is

$$y_{1k}(t) = \dot{y}_{1r}(t)\left(t - t_2\right) + x_d(t_2). \tag{24}$$

By repeating these procedures above, time t_i corresponding to the characteristic point in the region of multimodel for formation can be obtained, and at the same time the characteristic state point corresponding to the time t_i can be also obtained:

$$\begin{aligned} &\omega_w^i = 0, \qquad \varphi_w^i = \varphi_l, \qquad v_w^i = v_l, \\ &x_d^i = x_d(t_i), \qquad y_d^i = y_d(t_i). \end{aligned} \tag{25}$$

Calculate until the last characteristic state point is obtained, and then the computation will be terminated.

3.2. Generation of Discrete Model Sets for Formation Control. After obtaining the characteristic state points, carry on linearization at different discrete model sets of formation. Here, linearization can be realized through the following methods.

Consider nonlinear systems as described in the form of discrete-time dynamic equations:

$$\begin{aligned} x(k+1) &= f(x(k), u(k)), \\ y(k) &= g(x(k), u(k)). \end{aligned} \tag{26}$$

The system has m different characteristic state points. $f(x(k), u(k))$ and $g(x(k), u(k))$ have the first continuous partial derivative. If system is linearized at each characteristic state point, the standard discrete state-space model of m linear models of the original system is obtained as follows:

$$\begin{aligned} x(k+1) &= A_i x(k) + B_i u(k) - \alpha_i. \\ y(k) &= C_i x(k) + D_i u(k) - \beta_i, \end{aligned} \tag{27}$$

where

$$\begin{aligned} A_i &= \left.\frac{\partial f}{\partial x}\right|_{(x_i,u_i)}, \qquad B_i = \left.\frac{\partial f}{\partial u}\right|_{(x_i,u_i)}, \qquad C_i = \left.\frac{\partial g}{\partial x}\right|_{(x_i,u_i)}, \\ D_i &= \left.\frac{\partial g}{\partial u}\right|_{(x_i,u_i)}, \qquad \alpha_i = \left.\frac{\partial f}{\partial x}\right|_{(x_i,u_i)} x_i + \left.\frac{\partial f}{\partial u}\right|_{(x_i,u_i)} u_i - x_i, \\ \alpha_i &= \left.\frac{\partial g}{\partial x}\right|_{(x_i,u_i)} x_i + \left.\frac{\partial g}{\partial u}\right|_{(x_i,u_i)} u_i - y_i. \end{aligned} \tag{28}$$

Here, m linearized models constitute the linearized multimodel presentation of the original system.

For the UAV formation flying control, the characteristic state points are shown as

$$\begin{aligned} &\omega_w^i = 0, \qquad \varphi_w^i = \varphi_l, \qquad v_w^i = v_l, \\ &x_d^i = x_d(t_i), \qquad y_d^i = y_d(t_i). \end{aligned} \tag{29}$$

And it has the following expression:

$$\begin{aligned} A_i &= \left.\frac{\partial f}{\partial x}\right|_{(x_i,u_i)} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & -v_l(k)\Delta T \\ 0 & 0 & 1 \end{bmatrix}, \\ B_i &= \left.\frac{\partial f}{\partial u}\right|_{(x_i,u_i)} = \begin{bmatrix} -1 & y_d(t_i) \\ 0 & -x_d(t_i) \\ 0 & 1 \end{bmatrix} \Delta T, \\ \alpha_i &= \left.\frac{\partial f}{\partial x}\right|_{(x_i,u_i)} x_i + \left.\frac{\partial f}{\partial u}\right|_{(x_i,u_i)} u_i - x_i = \begin{bmatrix} -v_l(k)\Delta T \\ -v_l(k)\varphi_l(k)\Delta T \\ 0 \end{bmatrix}. \end{aligned} \tag{30}$$

Since outputs are linear, C_i, D_i and β_i will not be solved. Thus, the linearization equation at characteristic state point will be

$$\begin{aligned} &\begin{bmatrix} x_d(k+1) \\ y_d(k+1) \\ \varphi_d(k+1) \end{bmatrix} \\ &= \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & -v_l(k)\Delta T \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x_d(k) \\ y_d(k) \\ \varphi_d(k) \end{bmatrix} \\ &\quad + \begin{bmatrix} -1 & y_d(t_i) \\ 0 & x_d(t_i) \\ 0 & 1 \end{bmatrix} \Delta T \begin{bmatrix} v_w(k) \\ \omega_w(k) \end{bmatrix} - \begin{bmatrix} -v_l(k)\Delta T \\ -v_l(k)\varphi_l(k)\Delta T \\ 0 \end{bmatrix}. \end{aligned} \tag{31}$$

So for different characteristic state points, the linear model for UAV formation control at different horizons can be obtained, realizing the acquisition of model sets for UAV formation. And these models are denoted as $M(1), M(2), \ldots, M(S)$.

3.3. Combined Method of Model Sets for Formation Flight Control. The characteristic state points are obtained using the method above. When the error reaches the maximum value, switch to the new model, which ensures the maximum error value between the predictive trajectory and the reference trajectory. Thus, the determination of model region can be realized. It can be seen that the applicable range of desired model is divided based on the time region, so different sampling points have different models. But the predictive control is based on the future time region. So in this paper, the applicable model of predictive point is judged by the time region, and then this model is used to calculate the predictive value. The judgment rules of predictive model are described as follows.

Assuming that the time corresponding to the state characteristic points is $t_1 < t_2 < \cdots < t_{s-1} < t_s$ and the predictive horizon is $[t, t+N]$, then there will be the following.

If $[t, t+N] \in [t_i, t_{i+1}]$, then the final predictive model of all points is $M_T = M_i$; if $[t, t+N] \in [t_i, t_{i+N}]$, one can judge the interval $[t_h, t_{h+1}]$ of all points between prediction point $t+1$ to $t+N$ in sequence and denote the model of this point as $M_T = M_h$; if $[t, t+N] \in [t_s, \infty)$, the final predictive model of all the points will be $M_T = M_s$.

Based on this method, we can obtain the predictive function during the future horizon to determine the optimization index. However, the boundary point of the predictive range may be closer to the next linear model, as shown in Figure 4. Sampling point p_1 may be closer to the linear model at sampling point t_4. But according to the method above, the calculation model used at the sampling point p_1 is the linear model of the state characteristic point at sampling point t_3. Sampling point p_2 may be closer to the linear model of the state characteristic point at sampling point t_5. But according to the method above, the calculation model used by the sampling point p_2 is the linear model of the state characteristic point at sampling point t_4.

So the method above may decrease the performance of the approximation capability on the boundary, and each model belonging to the model set cannot switch smoothly [18]. However, T-S fuzzy model as an intelligent control method, mainly uses fuzzy reasoning to approximate the nonlinear system. Using this method, the input space can be divided into several fuzzy subspaces, where a local linear model is established, and then the local models are combined smoothly using the membership function, forming a global fuzzy model of nonlinear function, which is ultimately identified as a linear model [19]. The predictive control method based on the T-S fuzzy model belongs to multimodel predictive control with the weighted models. Compared with the common multimodel predictive controllers with weighted models, the fuzzy weighted models have more accurate nonlinear approximation performance; switch of

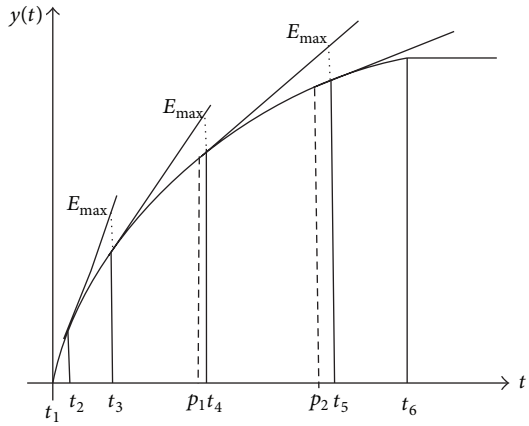


Figure 4: The schematic for determining model of frontier points during the predictive intervals.

the model is more smooth, and it is easier to understand [20, 21]. So in this section, T-S fuzzy idea is adopted for the multimodel control of UAV formation flight, as is shown in Figure 5.

For each sampling point, use the error between the tangent of state characteristic point and the reference trajectory of this sampling point to calculate the membership degree. Assuming that the error between tangent of jth state characteristic point and the reference trajectory at the sampling point t is $E_i(t)$, so for the point $t+i$ in the predictive range, weighted function is as follows:

$$\begin{aligned} w_j(t+i) &= e^{-(E_i(t+i)/E_{\max})^2}, \quad -E_{\max} \le E_j(t+i) \le E_{\max}, \\ w(t+i) &= 0, \qquad E_j(t+i) \le -E_{\max} \\ &\text{or} \quad E_j(t+i) \ge E_{\max}. \end{aligned} \tag{32}$$

This equation can ensure that the farther away from the state characteristic point the sampling point is, the lower its weighted value is. Using the weighted values, the prediction model at the sampling point t is

$$M_t = \frac{\sum_{i=1}^{m}(w_i M_i)}{\sum_{i=1}^{m}(w_i)}. \tag{33}$$

For the sampling points during the predictive horizon, there is

$$y_p(t+i) = \frac{\sum_{j=1}^{m}\left(w_j(t+i)\, M_j\right)}{\sum_{j=1}^{m}\left(w_j(t+i)\right)}, \tag{34}$$

where

$$\begin{aligned} w_j(t+i) &= e^{-(E_i(t+i)/E_{\max})^2}, \\ -E_{\max} &\le E_j(t+i) \le E_{\max}, \\ w(t+i) &= 0, \qquad E_j(t+i) \le -E_{\max} \\ &\text{or} \quad E_j(t+i) \ge E_{\max}. \end{aligned} \tag{35}$$

Through this approach, the linear prediction function for UAV formation can be obtained as follows:

$$\begin{aligned} &x_{dp}(k+1), x_{dp}(k+2), \ldots, x_{dp}(k+N), \\ &y_{dp}(k+1), y_{dp}(k+2), \ldots, y_{dp}(k+N). \end{aligned} \tag{36}$$

In this way, predictive outputs can change from a nonlinear function to a linear function. This nonlinear function includes $v_w(k), v_w(k+1), \ldots, v_w(k+N-1)$, and $\omega_w(k), \omega_w(k+1), \ldots, \omega_w(k+N-1)$, while this linear function includes the control quantities mentioned above. Thus, the control problem will become a multi-input-multioutput linear predictive control problem.

3.4. Optimization Index for Formation and Receding Optimization Solution. During the predictive control process, the goal of receding optimization is to find a set of $v_w(k), v_w(k+1), \ldots, v_w(k+N-1)$ and $\omega_w(k), \omega_w(k+1), \ldots, \omega_w(k+N-1)$, making prediction outputs at entire optimization horizon as close to the reference trajectory as possible.

Here, introduce the closed-loop

$$\begin{aligned} e_1(t) &= x_d(t) - x_{dp}(t), \\ e_2(t) &= y_d(t) - y_{dp}(t). \end{aligned} \tag{37}$$

The open-loop predictive output can be directly compensated by the output feedback, and then the predictive value of the closed-loop model will be

$$\begin{aligned} x_d(t+i) &= x_{dp}(t+i) + e_1(t), \\ y_d(t+i) &= y_{dp}(t+i) + e_2(t). \end{aligned} \tag{38}$$

In this section, there are two control objectives: the relative distances to X-axis and Y-axis. Since they have equal importance and the same unit of quantity, they are set with the same weight when designing the performance index. Thus, the performance index is defined as follows:

$$\begin{aligned} J = \sum_{i=1}^{N} &\left[\left(y_{1r}(t+i) - x_d(t+i)\right)^2\right. \\ &\left.+\left(y_{2r}(t+i) - y_d(t+i)\right)^2\right] \\ &+ \lambda_1 \sum_{i=0}^{N-1}\left(v_w(t+i) - v_w(t)\right)^2 \\ &+ \lambda_2 \sum_{i=0}^{N-1} \omega_w^2(t+i). \end{aligned} \tag{39}$$

Similarly, the optimization constraints of control quantity are introduced as follows:

$$\begin{aligned} v_w(k+i-1) - \Delta v < v_w(k+i) &< v_w(k+i-1) + \Delta v, \\ \omega_{\min} < \omega_w(k+i) &< \omega_{\max}, \\ v_{\min} < v_w(k+i) &< v_{\max}, \\ \text{where } i &\in 0, 1, 2, \ldots, N-1. \end{aligned} \tag{40}$$

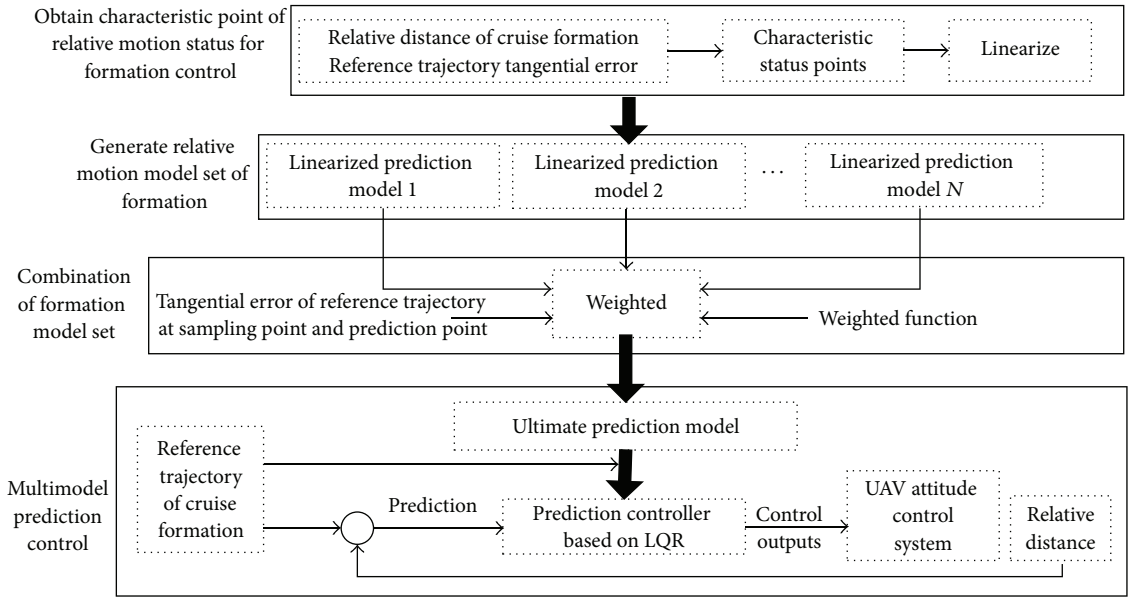


Figure 5: The schematic of multimodel control method for UAV formation flight.

After using multiple models, the performance index is linear quadratic, whose constraints are linear equality and inequality, so the optimization problem is a linear quadratic programming problem. The solution methods of linear quadratic programming problem can be used to solve the receding optimization problem. The linear quadratic programming problem is a common programming problem and has a lot of solution methods and higher speed than the ordinary nonlinear programming, which increases the speed of receding optimization solution [22].

4. Simulation

In this section, numerical simulations are performed to demonstrate the performance of the proposed approach. Here, the formation control ability can be tested in two important scenarios. Simulation scenarios are set as follows. One scenario is the leader UAV flying straight, and the other is the leader UAV flying with turning course. Additionally, the comparison simulation between single MPC (SMPC) method and multiple MPC (MMPC) method is carried on to verify effectiveness of the method in this paper. Meanwhile, the parameters used in the simulations are set as follows. The prediction horizon N is 5 and the sampling interval is 0.2 s. The angular velocity and velocity of two vehicles are confined during the interval $(-0.1, 0.1)$ and the interval $(35, 45)$, respectively. All the computations and experiments have been on a computer with Inter Core i3 CPU, 3.30 GHz, and Windows XP operating systems. Table 1 summarizes the initial conditions of the formation.

4.1. Formation Simulation of Leader UAV Flying Straight. The simulation experiment is mainly used to verify the UAV formation control capability when the leader UAV is flying straight. Here, error exists in the position measurement of leader UAV, which is ±0.5 m. There are two different control goals. One is that the relative position between follower and leader of UAV formation in the track coordinates is as follows:

$$X = -60, \quad Y = 30. \tag{41}$$

The other is that the formation should be formed within 40 s.

Because the leader UAV has its initial angle of $\pi/4$ and it flies straight, the initial relative position in the track coordinates will be obtained as follows:

$$\begin{bmatrix} x_d \\ y_d \end{bmatrix} = \begin{bmatrix} \cos\frac{\pi}{4} & \sin\frac{\pi}{4} \\ -\sin\frac{\pi}{4} & \cos\frac{\pi}{4} \end{bmatrix} \begin{bmatrix} -100 \\ -100 \end{bmatrix} = \begin{bmatrix} -100\sqrt{2} \\ 0 \end{bmatrix}. \tag{42}$$

Simulation is carried out by using Matlab Simulink toolbox, and the simulation results are shown from Figures 6, 7, 8, 9, and 10.

According to Figures 6–10, it can be seen that when leader UAV is navigating in a straight line, formation control can be achieved through both SMPC and MMPC method. However, the SMPC method has a larger tracking error than the MMPC method. Meanwhile, it can also be seen that it takes a longer time for SMPC method than MMPC method to

Table 1: Initial conditions of UAV formation.

Initial conditions	The role of UAV	
	Leader	Follower
Initial position	(0, 0)	(−100, −100)
Initial angle	0	$\pi/2$
Initial velocity	40	40
Initial angular velocity	0	0

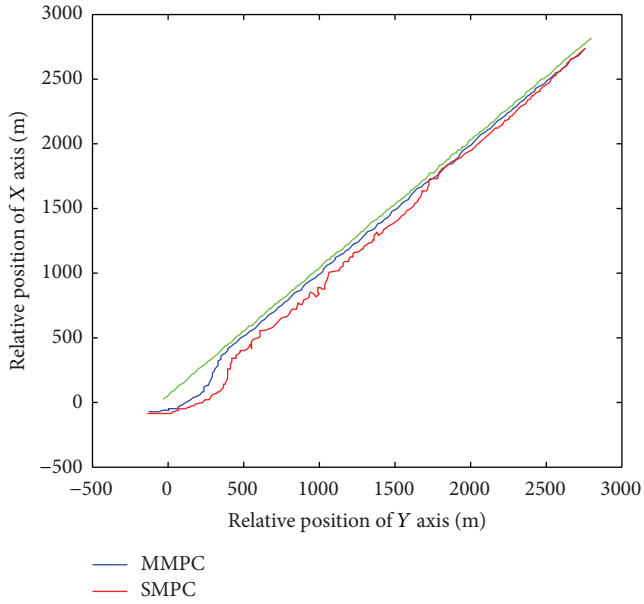


Figure 6: Flight trajectories of two UAVs.

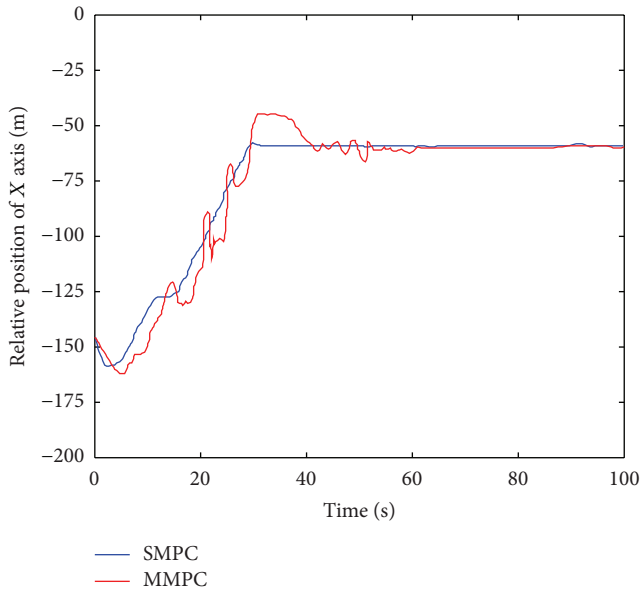


Figure 7: Relative position of X-axis.

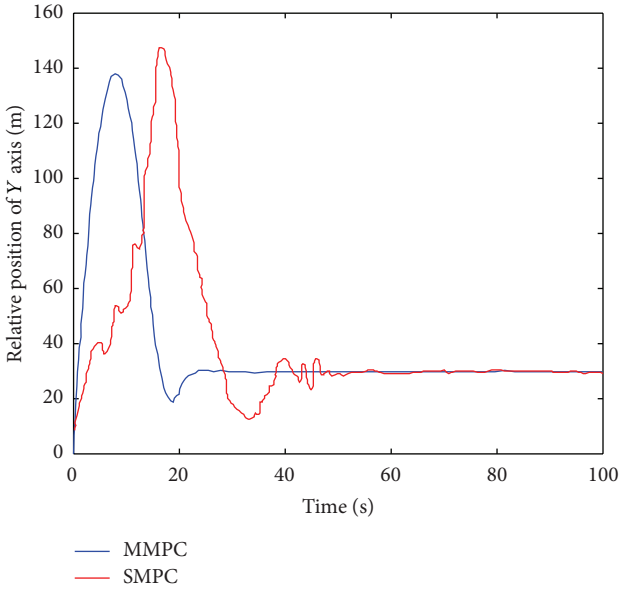


Figure 8: Relative position of Y-axis.

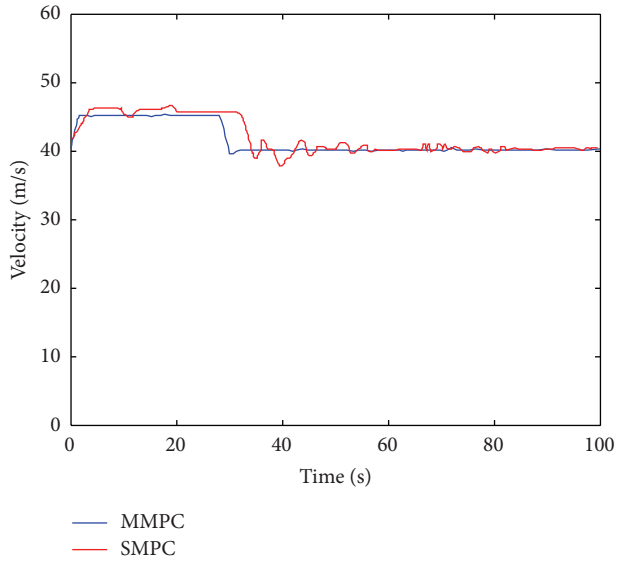


Figure 9: Change curve of follower UAV's velocity.

form a steady formation. The UAV formation can be realized in 40 seconds by the MMPC method, which meets the actual design demand.

4.2. Formation Simulation for Leader UAV with Turning Flight. The UAV formation control capability is proved in this section when the leader UAV flies with a turning flight path. The UAV flies 20 s with an initial angle of 0° between the leader UAV and X-axis, and then the UAV flies with angular velocity of $\pi/200$ for 100 seconds, and then it moves straight in Y-axis direction. There are also two different control goals. One is that the relative position between follower and leader of UAV formation in the track coordinates is as follows:

$$\begin{bmatrix} x_{\text{dref}} \\ y_{\text{dref}} \end{bmatrix} = \begin{bmatrix} -50 \\ -50 \end{bmatrix}. \tag{43}$$

The other is the formation should form within 40 s.

From Table 1, the relative position in the track coordinate system between two vehicles is obtained as follows:

$$\begin{bmatrix} x_d \\ y_d \end{bmatrix} = \begin{bmatrix} \cos 0 & \sin 0 \\ -\sin 0 & \cos 0 \end{bmatrix} \begin{bmatrix} -100 \\ -100 \end{bmatrix} = \begin{bmatrix} -100 \\ -100 \end{bmatrix}. \tag{44}$$

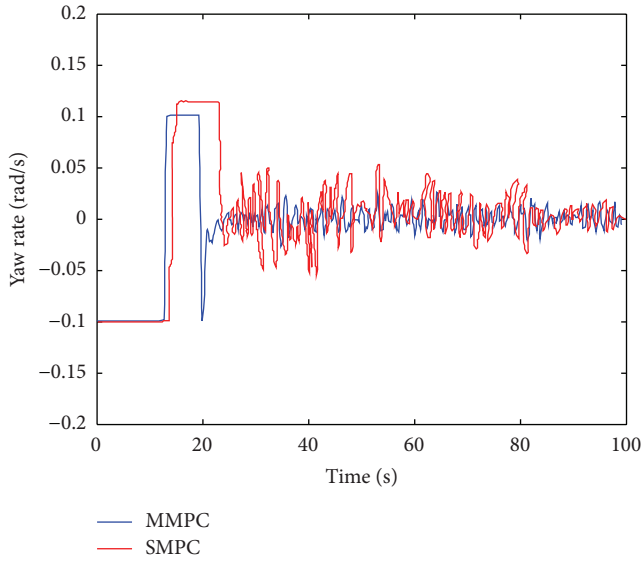


Figure 10: Change curve of follower UAV's yaw rate.

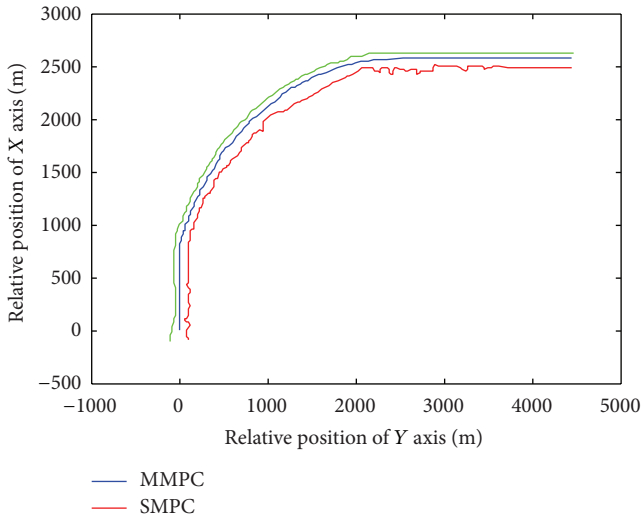


Figure 11: Flight trajectories of two UAVs.

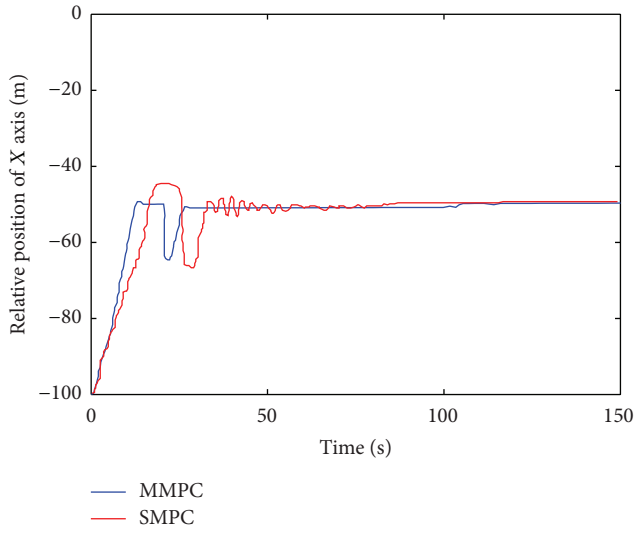


Figure 12: Relative position of X-axis.

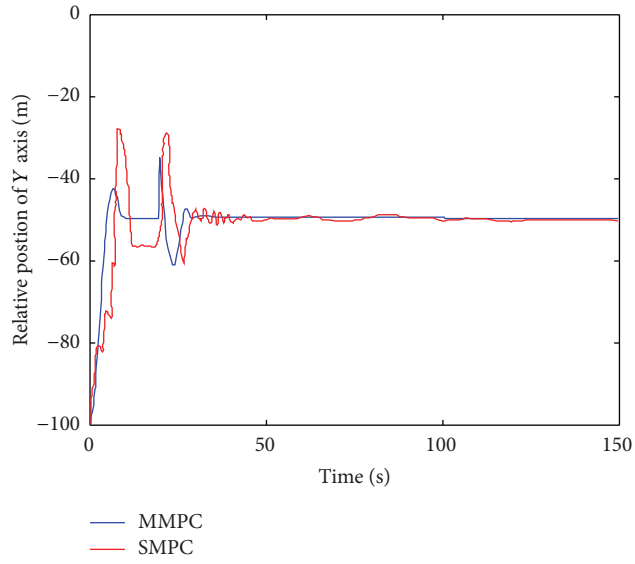


Figure 13: Relative position of Y-axis.

Simulation is carried out by using Matlab Simulink toolbox, and the results are shown in Figures 11, 12, 13, 14, and 15.

According to Figures 10–14, when leader UAV flies with a turning flight path, using method proposed in the paper, formation control can be achieved better than the SMPC method, whenever the UAV flies straight or flies with a turning path. The SMPC method has a larger tracking error than the MMPC method. Meanwhile, it can also be seen that it takes a longer time for SMPC method than MMPC method to form a steady formation. The UAV formation can be realized in 40 seconds by the MMPC method, which meets the actual design demand.

According to the Matlab simulation process of UAV formation in those two scenarios above, when the sampling interval is 0.2 s, the simulation time of the receding optimization program on the PC is less than 0.2 s each time and the time will be shorter if the simulation is done on a dedicated chip. So it meets the real-time needs. It can be seen from the relative position on the X-axis and Y-axis of two vehicles in the track coordinate system that the UAV formation is realized within 40 s. All in all, the simulation shows that the control requirements and real-time requirements can be satisfied by using multimodel predictive control method for UAV formation control.

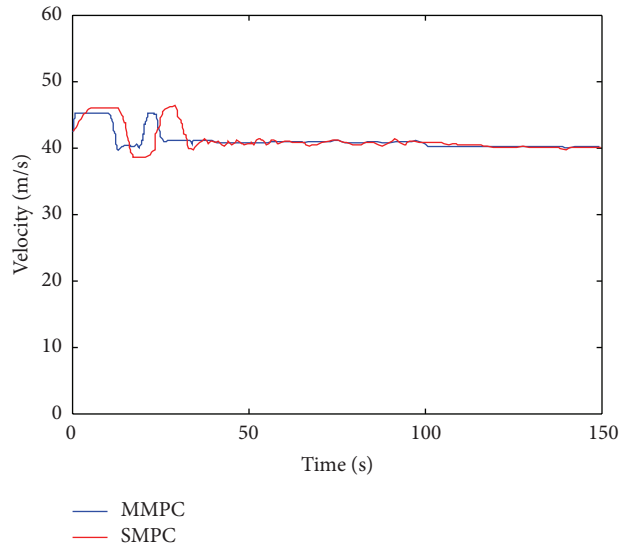


Figure 14: Change curve of follower UAV's velocity.

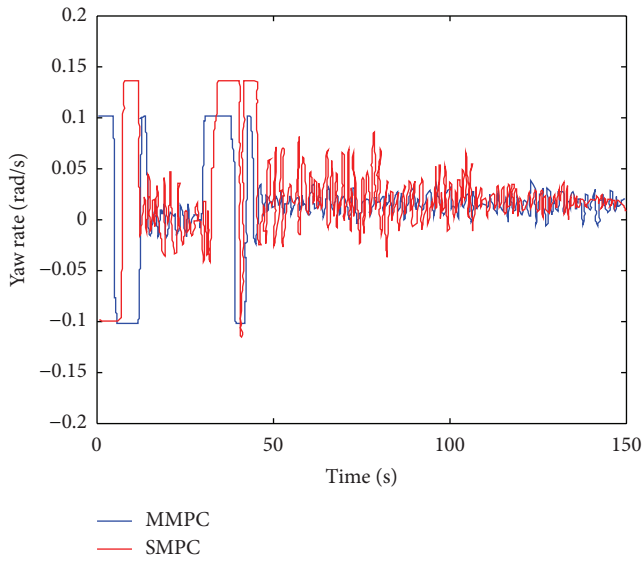


Figure 15: Change curve of follower UAV's yaw rate.

5. Conclusion

In this paper, the main work can be concluded as follows to solve the problem of UAV formation control.

(1) Discrete relative motion equations are established for UAV formation by using the leader-follower method.

(2) Multimodel sets for UAV formation are established, and the weighted model sets method is proposed.

(3) The formation controller based on multimodel predictive control is designed.

(4) Simulation in two scenarios is carried out and the effectiveness of controller designed and control strategy is verified.

The multimodel predictive control method can be used for UAV formation control. This method can meet control requirements and real-time requirements well. The result of this paper is the basis of further research on formation reconfiguration control problem. In the future, we will introduce the approach proposed in this paper to the controller design of actual UAV formation flight.

Conflict of Interests

The authors declare that there is no conflict of interests regarding the publication of this paper.

Acknowledgment

Research is supported by the National Science Foundation of China (NSFC) under Grants no. 51201182 and 61105012, and National Aviation Science Foundation of China (NASFC) under Grant no. 20135896027. Among these foundations, the NASFC is a cooperation program of our research group and FACRI, and this foundation requires both sides to publish an article.

References

[1] X.-Y. Wang, X.-M. Wang, and C.-C. Yao, "Design of UAVs formation flight controller based on neural network adaptive inversion," *Control and Decision*, vol. 28, no. 6, pp. 837–843, 2013.

[2] C.-J. Ru, R.-X. Wei, J. Dai, D. Shen, and L.-P. Zhang, "Autonomous reconfiguration control method for UAV's formation based on Nash bargain," *Auta Automatica Sinica*, vol. 39, no. 8, pp. 1349–1359, 2013.

[3] L. Jieun, S. K. Hyeong, and K. Youdan, "Formation geometry center based formation controller design using Lyapunov stability theory," *KSAS International Journal*, no. 2, pp. 71–76, 2008.

[4] A. Bemporad and C. Rocchi, "Decentralized hybrid model predictive control of a formation of unmanned aerial vehicles," in *Proceedings of the 18th IFAC Word Congress*, Milanno, Italy, 2011.

[5] Z. Chao, S.-L. Zhou, L. Ming, and W.-G. Zhang, "UAV formation flight based on nonlinear model predictive control," *Mathematical Problems in Engineering*, vol. 2012, Article ID 261367, 15 pages, 2012.

[6] K. Wesselowski and R. Fierro, "A dual-mode model predictive controller for robot formations," in *Proceedings of the 42nd IEEE Conference on Decision and Control*, pp. 3615–3620, Maui, Hawaii, USA, December 2003.

[7] B. J. N. Guerreiro, C. Silvestre, and R. Cunha, "Terrain avoidance nonlinear model predictive control for autonomous rotorcraft," *Journal of Intelligent & Robotic Systems: Theory and Applications*, vol. 68, no. 9, pp. 69–85, 2012.

[8] M. A. Abbas, J. M. Eklund, and R. Milman, "Real-time analysis for nonlinear model predictive control of autonomous vehicles," in *Proceedings of the 25th IEEE Canadian Conference on Electrical & Computer Engineering (CCECE '12)*, pp. 1–4, 2012.

[9] J. Shin and H. J. Kim, "Nonlinear model predictive formation flight," *IEEE Transactions on Systems, Man, and Cybernetics A: Systems and Humans*, vol. 39, no. 5, pp. 1116–1125, 2009.

[10] F. Alessandro, L. Sauro, and M. Andrea, "Nonlinear decentralized model predictive control strategy for a formation of unmanned aerial vehicles," in *Proceedings of the 2nd IFAC Workshop on Multivehicle System*, vol. 2, pp. 49–54, 2012.

[11] C. Gorman and N. Slegers, "Predictive control of general nonlinear systems using series approximations," in *Proceedings of the AIAA Guidance, Navigation, and Control Conference and Exhibit*, AIAA 2009-5994, Chicago, Ill, USA, August 2009.

[12] W. Dhouib, M. Djemel, and M. Chtourou, "Fuzzy predictive control of nonlinear systems," in *Proceedings of the 8th International Multi-Conference on Systems, Signals and Devices (SSD '11)*, pp. 1–8, Sousse, Tunisia, March 2011.

[13] T. Keviczky, F. Borrelli, and G. J. Balas, "Decentralized receding horizon control for large scale dynamically decoupled systems," *Automatica*, vol. 42, no. 12, pp. 2105–2115, 2006.

[14] Q. Chen, L. Gao, R. A. Dougal, and S. Quan, "Multiple model predictive control for a hybrid proton exchange membrane fuel cell system," *Journal of Power Sources*, vol. 191, no. 2, pp. 473–482, 2009.

[15] N. N. Nandola and S. Bhartiya, "A multiple model approach for predictive control of nonlinear hybrid systems," *Journal of Process Control*, vol. 18, no. 2, pp. 131–148, 2008.

[16] D. Dougherty and D. Cooper, "A practical multiple model adaptive strategy for single-loop MPC," *Control Engineering Practice*, vol. 11, no. 2, pp. 141–159, 2003.

[17] K. S. Narendra and C. Xiang, "Adaptive control of discrete-time systems using multiple models," *IEEE Transactions on Automatic Control*, vol. 45, no. 9, pp. 1669–1686, 2000.

[18] L.-L. Liu, L.-F. Zhou, T. Ji, and Y.-H. Zhao, "Research on model switching method of multi-hierarchical model predictive control systems," *Acta Automatica Sinica*, vol. 39, no. 5, pp. 626–630, 2013.

[19] R. J. Spiegel, M. W. Turner, and V. E. McCormick, "Fuzzy-logic-based controllers for efficiency optimization of inverter-fed induction motor drives," *Fuzzy Sets and Systems*, vol. 137, no. 3, pp. 387–401, 2003.

[20] Z.-Q. Chen and H.-M. Jiang, "T-S fuzzy model predictive control simulation based on intelligent optimization algorithm," *Journal of System Simulation*, vol. 2, pp. 79–85, 2009.

[21] Y. Gu, H. O. Wang, K. Tanaka, and L. G. Bushnell, "Fuzzy control of nonlinear time-delay systems: stability and design issues," in *Proceedings of the American Control Conference (ACC '01)*, pp. 4771–4776, Arlington, Calif, USA, June 2001.

[22] T. Keviczky, F. Borrelli, K. Fregene, D. Godbole, and G. J. Balas, "Decentralized receding horizon control and coordination of autonomous vehicle formations," *IEEE Transactions on Control Systems Technology*, vol. 16, no. 1, pp. 19–33, 2008.

Hindawi Publishing Corporation
Mathematical Problems in Engineering
Volume 2014, Article ID 368961, 12 pages
http://dx.doi.org/10.1155/2014/368961

Research Article

Localization of Outdoor Mobile Robots Using Curb Features in Urban Road Environments

Hyunsuk Lee,[1] Jooyoung Park,[2] and Woojin Chung[1]

[1] *Department of Mechanical Engineering, Korea University, Anam-dong, Seongbuk-gu, Seoul 136-713, Republic of Korea*
[2] *Department of Control and Instrumentation, Korea University, Anam-dong, Seongbuk-gu, Seoul 136-713, Republic of Korea*

Correspondence should be addressed to Woojin Chung; smartrobot@korea.ac.kr

Received 12 December 2013; Revised 13 February 2014; Accepted 13 February 2014; Published 8 April 2014

Academic Editor: Leo Chen

Urban road environments that have pavement and curb are characterized as semistructured road environments. In semistructured road environments, the curb provides useful information for robot navigation. In this paper, we present a practical localization method for outdoor mobile robots using the curb features in semistructured road environments. The curb features are especially useful in urban environment, where the GPS failures take place frequently. A curb extraction is conducted on the basis of the Kernel Fisher Discriminant Analysis (KFDA) to minimize false detection. We adopt the Extended Kalman Filter (EKF) to combine the curb information with odometry and Differential Global Positioning System (DGPS). The uncertainty models for the sensors are quantitatively analyzed to provide a practical solution.

1. Introduction

Outdoor environments have irregular shapes and changes in geometry and illumination due to the weather condition. Therefore, environmental uncertainty is relatively high. There are numerous studies for autonomous navigation of mobile robot in outdoor environments. Typical examples are the autonomous vehicles that were developed through DARPA Grand/Urban Challenges [1, 2]. Most of the vehicles were equipped with a variety of high cost sensors in order to overcome various uncertainties.

The aim of this work is to develop a practical localization method for outdoor mobile robots. In particular, this study focuses on surveillance robots in urban road environments. A localization method using a small number of sensors is proposed instead of using multiple high cost sensors.

The fusion of a global positioning system (GPS) and inertial measurement unit (IMU) has been widely used for the outdoor localization of mobile robots [3, 4]. However, it is difficult to ensure the accurate pose estimation in dense urban environment, where GPS signal is degraded by the multipath errors and satellite blockage. Therefore, the use of environmental features has been studied to enhance the precision of the estimated robot pose in dense urban environment [5, 6]. In [7], a 3D representation of the local environment is used to detect obstruction of the GPS signals, which is blocked by buildings. In [8, 9], the extracted line features of buildings are used to estimate the robot position. However, the available information regarding buildings is sparse in many places, and the slow update rate limits the performance of localization. In [10], the road centerline is extracted to correct the lateral position of the vehicle in mountainous forested paths. However, the correction of the heading error is not considered by the extracted road centerline.

Generally, urban road environments are paved, and the curbs act as the boundaries of the roads. Therefore, urban road environments are characterized as semistructured road environments. In semistructured road environments, the curb provides useful information for robot navigation. Therefore, the curb features have been widely used for navigation strategies and localization methods. In [11], Wijesoma et al. propose a method based on EKF for detection and tracking of the curbs. The range and angle of the curbs are obtained from an LRF measurement. However, quantitative performance analysis of the curb detection was not clearly shown. In [12, 13], the curb on one side of the road is extracted using a vertical LRF for vehicle localization. The lateral error of

a vehicle is reduced by map matching approach using the extracted curb point. However, the correction of the heading angle is not considered, because there is no angle information of the curb.

A method for traversable region detection using road features such as road surface, curbs, and obstacles is proposed in our previous works [14–16]. The curb features are derived by the geometric features of the road to obtain curb candidates. In order to extract the correct curb among the curb candidates, a validation gate is derived from principal component analysis (PCA). The curb extraction is performed successfully in a road environment. However, there was a fundamental limitation on PCA. The validation gate does not consider the classification of the curb and noncurb data. Consequently, false detection where noncurb data are extracted as curbs may occur. The precision of estimated pose is decreased, when the false detection data are used for localization. Therefore, it is important to reduce the false detection rate.

The contribution of this paper can be summarized by two schemes. The first contribution is the robust curb extraction scheme by using a single Laser Range Finder (LRF). In order to reduce the number of false detection of the curbs, the classification of the curb data and the noncurb data is conducted by using Kernel Fisher Discriminant Analysis (KFDA) in [17–19]. On the basis of our previous works, geometrical features of the curb are defined. The second contribution is the integrated localization scheme using curb features on the basis of quantitative sensor uncertainty models. In particular, the uncertainty model for the extracted curb is quantitatively determined from experiments. Extended Kalman Filter (EKF) is exploited to combine the curb features with odometry and Differential Global Positioning System (DGPS) information. The extracted curbs enable accurate estimation of the pose of the robot even when the temporal GPS blackout takes place.

The remainder of this paper is organized as follows. A method for curb extraction is presented in Section 2. Section 3 describes the localization method for an outdoor mobile robot using the extracted curb. It also introduces the uncertainty models for the sensor measurements. The experimental results of the proposed method are presented in Section 4. Finally, the conclusion of this research is presented in Section 5.

2. Road Feature Extraction

2.1. System Configuration. Figure 1 shows the configuration of the mobile robot and the LRF coordinate configuration. A single onboard LRF with a tilting angle of α_t is used to extract road features such as the road surface and curbs. The following list shows the nomenclature for road feature extraction. The variables are described with respect to the robot coordinate frame:

w_r: road width;

α_t: nominal tilt angle of LRF;

θ_f: angle between the detected road surface points and Y_R-axis;

d_f: horizontal look-ahead distance from the robot to the road surface points;

$(x_{C,R}, y_{C,R})$: coordinates of the right curb edge (point C);

$\theta_{C,R}$: angle between the right curb and Y_R-axis;

$(x_{C,L}, y_{C,L})$: coordinates of the left curb edge (point B);

$\theta_{C,L}$: angle between the left curb and Y_R-axis.

2.2. Road Feature Detection. A road feature detection scheme was proposed in our previous works [14, 15]. The previous works on road feature detection are briefly reviewed. The first step for the road feature detection is the road surface extraction. In order to identify the road surface, the LRF measurements in expected road region R_y are selected as candidates for the road surface L_r. Multiple line segments are constructed by combining consecutive data points of L_r. The angle of the road surface is parallel with the X_R-axis in the ideal case. Therefore, the range data L_r are saved as road surface points L_f if the angle of the line segment is within threshold θ_e. The road surface is extracted as a line by using the least square method. The extracted road surface provides θ_f and d_f from the robot. The overall algorithm can be summarized in Algorithm 1.

Figure 2 shows the ideal model of a semistructured road environment. The geometrical features of the road can be used to extract the curb features. The curb has the following four attributes.

(Att_1) The angular difference between the curb orientation (θ_c) and road surface angle (θ_f) is close to 90°.

(Att_2) The gap between the horizon distance of the road surface (d_f) and the y value of the curb point (B or C) is close to 0.

(Att_3) The angular difference between the left curb ($\bar{AB}$) and the right curb ($\bar{CD}$) is close to 0.

(Att_4) The difference between the road width and the gap between two curbs is close to 0. It is assumed that the road width is known.

It is commonly assumed that the robot navigates parallel to the curb. It is reasonable to assume that the robot is moving along the road without significant change of orientation in most cases. Moreover, the vertical surfaces of the curb are perpendicular to the road surface. When we scan the road environments using the LRF, the road features are composed of straight lines with different orientations. In order to distinguish different line segments that correspond to the road surface, the curbs, and the sidewalk, it is helpful to assume that the robot's heading direction points forward along the road. The road width is assumed to be known by given map information.

The first attribute is used to select the curb candidates. The line segments that satisfy the following condition are selected as the curb candidates:

$$a_1 = \frac{\pi}{2} \pm \varepsilon, \tag{1}$$

(1) $L_i \leftarrow$ the LRF measurement.
(2) $L_{Y,i} \leftarrow y$ coordinate of the LRF measurement.
(3) $N \leftarrow$ The number of the LRF measurement points.
(4) $R_Y \leftarrow$ Expected y range of the road surface points
(5) **for** $i \leftarrow 1$ to N
(6) **if** $L_{Y,i} \in R_Y$, **then**
(7) $L_r \leftarrow$ save L_i as candidate for the road surface.
(8) $\theta_r \leftarrow$ angle of candidate for the road surface.
(9) $M \leftarrow$ The number of candidates for the road surface.
(10) **end if**
(11) **end for**
(12) **for** $j \leftarrow 1$ to M
(13) **if** $\|\theta_{r,j}\| \leq \theta_e$, **then**
(14) $L_f \leftarrow$ save $L_{r,j}$ as the road surface.
(15) **end if**
(16) **end for**
(17) **return** (L_f)

Algorithm 1: Road surface extraction.

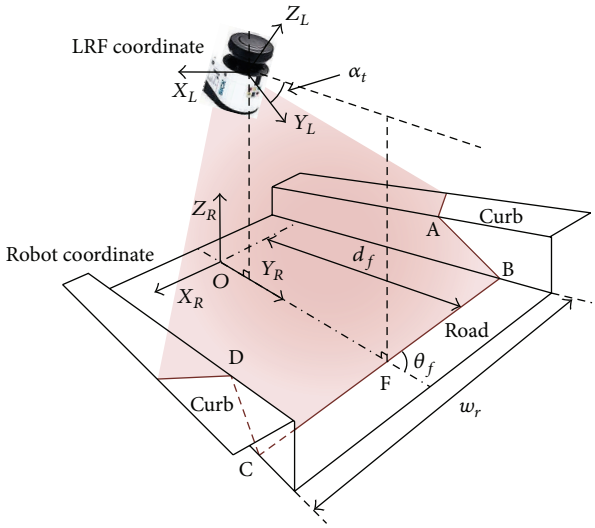


Figure 1: The configuration of a robot and installation of a LRF.

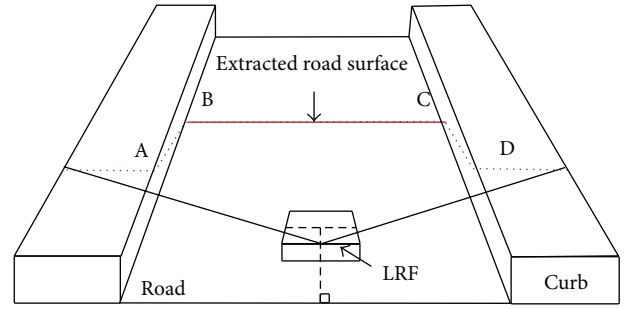


Figure 2: Ideal model of semiconstructed environment (blue dotted line: LRF data, red line: extracted road surface).

where $a_1 = |\theta_f - \theta_c|$, ε = tolerance.

Once the curb candidates are determined, attributes 2, 3, and 4 are used to extract the correct curb out of the curb candidates. The attribute values a_2, a_3, and a_4 that correspond to each attribute are numerically calculated for the pair of right and left curb candidates as follows:

$$\begin{aligned} a_2 &= d_f - \text{avg}\left(y_{C,R}, y_{C,L}\right), \\ a_3 &= \theta_{C,R} - \theta_{C,L}, \\ a_4 &= w_r - w_c; \end{aligned} \tag{2}$$

w_c is the gap between the right and left curb points ($w_c = x_{C,R} - x_{C,L}$). When a curb exists on only one side, a_2 and a_3 are computed from the curb candidates for one side. a_4 is assumed to be 0. The data vector of the curb candidates is given as follows:

$$\mathbf{x}_i = \begin{bmatrix} a_2 & a_3 & a_4 \end{bmatrix}^T. \tag{3}$$

2.3. Curb Extraction Using Kernel Fisher Discriminant Analysis (KFDA). KFDA is applied to extract the correct curb from the curb candidates. KFDA aims to find a discriminant function for optimal data classification. Therefore, the discriminant function can be used to classify the curb candidates as curb and noncurb class. The curb extraction is conducted by the following procedure. First, the training data with class information are selected. The discriminant function is derived from the offline computation of the training data. When the discriminant function is obtained, classification of the curb candidates is carried out by the discriminant function in real time.

The discriminant function α is defined as a vector to maximize the following object function in the kernel feature space $\mathcal{F}$. Consider the following:

$$J(\alpha) = \frac{\alpha^T S_B^\Phi \alpha}{\alpha^T S_W^\Phi \alpha}. \tag{4}$$

The "between-class variance matrix" S_B^Φ and the "within-class variance matrix" S_W^Φ are defined as follows:

$$\begin{aligned} S_B^\Phi &= \sum_c \ell_c \left(\kappa_c \kappa_c^T - \kappa\kappa^T\right), \\ S_W^\Phi &= KK^T - \sum_c \ell_c \kappa_c {\kappa_c}^T \end{aligned} \tag{5}$$

with $\kappa_c = (1/\ell_c)\sum_{i\in c} K_{ij}$ and $\kappa = (1/\ell)\sum_i K_{ij}$. ℓ is the number of total training samples, and ℓ_c is the number of class samples.

Training data $\{x_1, x_2, \ldots, x_\ell\}$ consist of data with obvious class information. Each sample $\mathbf{x}_{\text{training},i} \in \mathbb{R}^3$ belongs to one of two classes that include curb and noncurb class. In order to obtain equivalent effects on classification, the components of each sample should be normalized. Consider the following:

$$\mathbf{x}_i^{(k)} = \frac{\mathbf{x}_i^{(k)} - \mu^{(k)}}{\sigma^{(k)}}. \tag{6}$$

Here, $\mu^{(k)}$ and $\sigma^{(k)}$ are the mean and standard deviation of each attribute, respectively. The training data that are mapped to the kernel feature space are represented by an $\ell \times \ell$ matrix K. The Gaussian kernel is used for mapping the data onto kernel feature space $\mathcal{F}$. Consider the following:

$$K_{ij} = k\left(\mathbf{x}_i, \mathbf{x}_j\right) = \exp\left(-\frac{\left\|\mathbf{x}_i - \mathbf{x}_j\right\|^2}{2\sigma^2}\right). \tag{7}$$

σ is the control parameter that needs to be tuned to improve classification performance [19]. If σ is too small, the overfitting problem may take place. Although the classification accuracy with respect to the training data increases, the classification performance with respect to the test data becomes poor under the occurrence of the overfitting. If σ is too large, the underfitting problem may take place. The classification performance will not be satisfactory at all cases when the underfitting takes place. Therefore, σ should be carefully selected. In this paper, σ is manually tuned under the consideration of experimental performances.

The discriminant function that maximizes the object function in (4) is derived by an eigenvalue problem. In order to project the training data onto a one-dimensional (1D) solution space, the eigenvector that corresponds to the largest eigenvalue is defined as the discriminant function α for data classification. The discriminant function α is given by an $\ell \times 1$ vector. Consider the following:

$$S_B^\Phi \alpha = \lambda S_W^\Phi \alpha. \tag{8}$$

The classification of the training data is conducted by an inner product of the data matrix K and the discriminant function α. The projected training data y_{training} are distributed in the 1D solution space. Consider the following:

$$y_{\text{training}} = \alpha \cdot K. \tag{9}$$

The class of test data can be predicted by using the discriminant function α. The test data denote the curb candidates. The test data $\mathbf{x}_{\text{test}}$ is projected to the 1D solution space, as shown in the following equation:

$$y_{\text{test}} = \sum_i \alpha_i k\left(\mathbf{x}_{\text{training},i}, \mathbf{x}_{\text{test}}\right). \tag{10}$$

The class properties of the training data are used to predict the class of the test data y_{test}. The Mahalanobis distances between each class and the test data are computed by the following equation:

$$d\left(y_{\text{test}}, \mu_c^\Phi\right) = \frac{\left(y_{\text{test}} - \mu_c^\Phi\right)^2}{\left(\sigma_c^\Phi\right)^2}, \tag{11}$$

where μ_c^Φ and σ_c^Φ denote the mean and standard deviation of the class distribution, respectively. The class of the test data is determined as the class with the smallest Mahalanobis distance. When the test data are classified as the curb class, they are extracted as the curb.

3. Outdoor Localization Using Curb Feature

3.1. System Design. This paper adopts EKF to estimate the robot pose using curb features. EKF is a well-known method for mobile robot localization and sensor fusion [20–22]. When the initial pose of the robot and adequate observations are provided, the pose of the robot can be estimated by correcting the odometry error. The EKF process consists of a prediction step and an update step in sampling time k. DGPS and an LRF are used to correct odometry errors. Figure 3 shows a block diagram of the localization process.

3.2. Measurement Uncertainty Model. The GPS error mainly occurs due to the following two factors. One is the pseudorange errors caused by systematic factors. Another is the geometric constellation of satellites. In this paper, DGPS is used to minimize the pseudorange errors. The uncertainty model for DGPS is computed under the consideration of the "dilution of precision" in relation to the geometric constellation of satellites and the pseudorange error or so-called "user-equivalent range error" [23]. Consider the following:

$$\mathbf{R}_{\text{DGPS},k} = \begin{bmatrix} \sigma_x^2 & \sigma_{xy} \\ \sigma_{xy} & \sigma_y^2 \end{bmatrix} \sigma_{\text{UERE}}^2. \tag{12}$$

The error covariance $\mathbf{R}_{\text{DGPS},k}$ is given as (12). The elements of the covariance matrix are provided by the DGPS receiver in real time.

Several studies were proposed to define the error covariance of line features [24, 25]. However, the uncertainty models are appropriate for a static condition or a low-speed driving condition on a flat surface. However, the outdoor

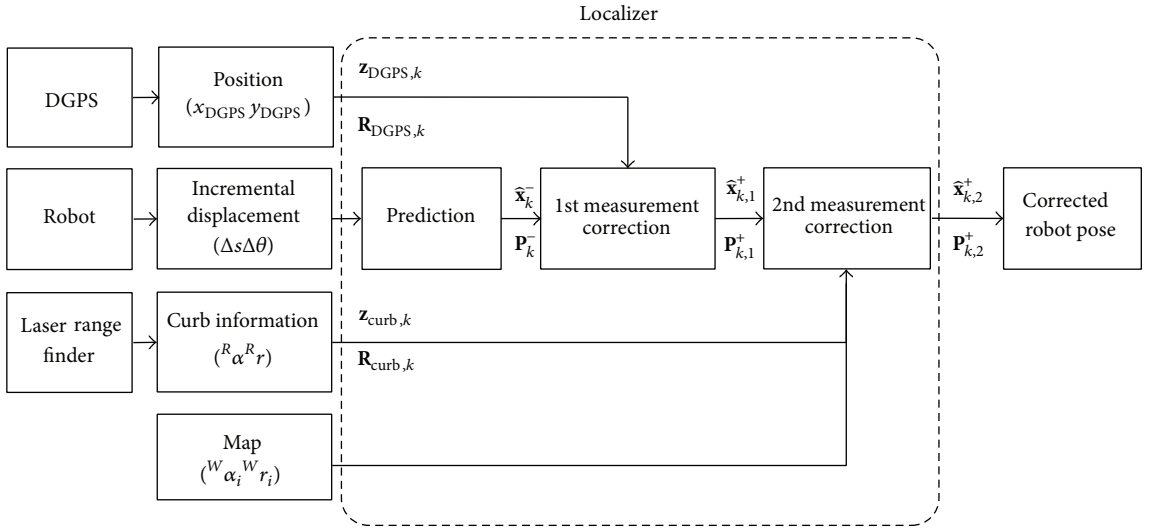


Figure 3: A block diagram of localization process.

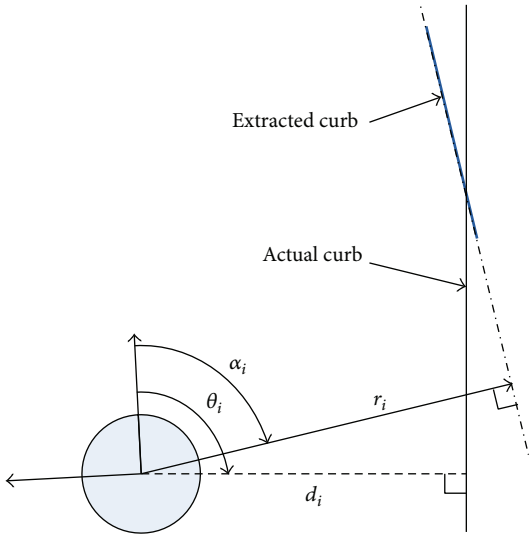


Figure 4: Noise model of the extracted curb.

mobile robots are usually driven in the uneven terrain. The measurement noise of the extracted curb occurs due to the wobble of the robot. Therefore, the uncertainty model for the curb needs to be defined by experiments under resultant driving conditions.

In order to define the error covariance $\mathbf{R}_{curb}$, we consider the noise model of the extracted curb as shown in Figure 4. The information of the extracted curb contains estimation errors in the range and angle. Therefore, the angle and range measurements for the extracted curb are composed of the "true" angle θ_i and the "true" range d_i, along with the estimation errors. Consider the following:

$$\begin{aligned} \alpha_i &= \theta_i + \varepsilon_{\alpha_i}, \\ r_i &= d_i + \varepsilon_{r_i}, \end{aligned} \tag{13}$$

where ε_{α_i} and ε_{r_i} are the estimation errors for a curb and have random variances σ_α^2 and σ_r^2, respectively. The ground truth of the curb locations can be measured by an additional LRF, which is attached to the side of the robot. Because the additional LRF is directly facing the curb, the range data from the additional LRF provide reliable geometric information of the curb during robot's movement. The measurement of the curb from the additional LRF is more accurate than the forward-pointing tilted LRF, because the number of range data points that correspond to the curb is much larger. The curb is represented as a straight line by the application of the least square method. The ground truth implies the relative range and orientation of the curb. The estimation errors are computed for the extracted curbs. Therefore, the error covariance of the curb is defined by the distribution of the estimation errors. By using a large number of measurements of the estimation errors ε_α and ε_r, the error covariance is calculated as follows:

$$\mathbf{R}_{\text{curb}} = \begin{bmatrix} \sigma_\alpha^2 & \sigma_{\alpha r} \\ \sigma_{r\alpha} & \sigma_r^2 \end{bmatrix} = \begin{bmatrix} E\left(\varepsilon_\alpha \varepsilon_\alpha^T\right) & E\left(\varepsilon_\alpha \varepsilon_r^T\right) \\ E\left(\varepsilon_r \varepsilon_\alpha^T\right) & E\left(\varepsilon_r \varepsilon_r^T\right) \end{bmatrix}. \tag{14}$$

The covariance matrices are experimentally defined as constant values for the left and right curbs.

3.3. Extended Kalman Filter Localization Using Curb Feature. The odometry data from the wheel encoders are used to predict the robot pose. By using the incremental distance Δs_{k-1} and orientation $\Delta\theta_{k-1}$, the predicted robot pose is given by the following equation:

$$\hat{\mathbf{x}}_k^- = f\left(\hat{\mathbf{x}}_{k-1}, \mathbf{u}_{k-1}\right) = \begin{bmatrix} \hat{x}_{k-1}^- + \Delta s_{k-1} \cos\left(\hat{\theta}_{k-1}^- + \dfrac{\Delta\theta_{k-1}}{2}\right) \\ \hat{y}_{k-1}^- + \Delta s_{k-1} \sin\left(\hat{\theta}_{k-1}^- + \dfrac{\Delta\theta_{k-1}}{2}\right) \\ \hat{\theta}_{k-1}^- + \Delta\theta_{k-1} \end{bmatrix}. \tag{15}$$

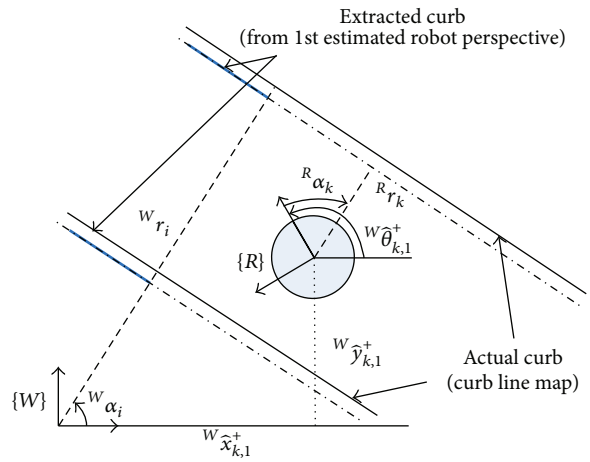


Figure 5: Extracted curb features and a line map with respect to the global coordinate frame.

The state vector $\hat{\mathbf{x}}_k^- = \begin{bmatrix}\hat{x}_k^- & \hat{y}_k^- & \hat{\theta}_k^-\end{bmatrix}^T$ is the predicted robot pose at sampling time k. When the initial pose of the robot is given, the robot pose is represented by x, y, and θ in a global coordinate frame. The uncertainty of the predicted robot pose consistently increases because of the accumulative errors of the odometry.

In order to correct the odometry error, the first measurement correction is performed using DGPS measurements. The update frequency is set to 1 Hz on the basis of the DGPS measurement frequency. When the available position measurement is provided, the observation vector is given in global coordinates. Consider the following:

$$\mathbf{z}_{\mathrm{DGPS},k} = \begin{bmatrix} x_{\mathrm{DGPS},k} & y_{\mathrm{DGPS},k}\end{bmatrix}^T. \tag{16}$$

The observation model for the current state is described as follows:

$$\mathbf{h}_{\mathrm{DGPS}}\left(\hat{\mathbf{x}}_k^-, k\right) = \begin{bmatrix}\hat{x}_k^- & \hat{y}_k^-\end{bmatrix}^T. \tag{17}$$

The measurement Jacobian matrix $\mathbf{H}_k$ is given as follows:

$$\mathbf{H}_{\mathrm{DGPS},k} = \frac{\partial \mathbf{h}\left(\hat{\mathbf{x}}_k^-, k\right)}{\partial \hat{\mathbf{x}}_k^-} = \begin{bmatrix}1 & 0 & 0\\ 0 & 1 & 0\end{bmatrix}, \tag{18}$$

$$\mathbf{z}_{\mathrm{curb},k} = \begin{bmatrix}\alpha_k & r_k\end{bmatrix}^T. \tag{19}$$

The second measurement correction is conducted by using the curb features. The observation vector for the extracted curb is given by (19). When the curbs are extracted on both sides, there are two measurement vectors for the left and right curbs as shown in Figure 5. The sequence of the second measurement correction is from the right curb to the left curb. The update rate is 5 Hz if the curbs are continuously extracted.

The robot pose is corrected by comparing the curb with the map. The extracted curb is matched with the ith line of the curb map. The observation model for the ith line $^{W}\mathbf{z}_i = \begin{bmatrix}^{W}\alpha_i & ^{W}r_i\end{bmatrix}^T$ and the robot pose at time k is given by the following equation:

$$\mathbf{h}_{\mathrm{curb}}\left(\hat{\mathbf{x}}_{k,1}^+, k, i\right) = \begin{bmatrix} ^{W}\alpha_i - \hat{\theta}_{k,1}^+ \\ ^{W}r_i - \left\{\hat{x}_{k,1}^+ \cos {}^{W}\alpha_i + \hat{y}_{k,1}^+ \sin {}^{W}\alpha_i\right\}\end{bmatrix}. \tag{20}$$

The Jacobian matrix $\mathbf{H}_k$ is defined as follows:

$$\mathbf{H}_{\mathrm{curb},k} = \frac{\partial \mathbf{h}\left(\hat{\mathbf{x}}_{k,1}^+, k, i\right)}{\partial \hat{\mathbf{x}}_{k,1}^+} = \begin{bmatrix}0 & 0 & -1\\ -\cos {}^{W}\alpha_i & -\sin {}^{W}\alpha_i & 0\end{bmatrix}. \tag{21}$$

The consistency of EKF relies on the observation model. If an erroneous sensor observation is provided, the system does not provide a consistent result. Therefore, the outliers that lie outside of the uncertainty bounds should be rejected. A normalized innovation squared (NIS) test is implemented in order to confirm the consistency of the filter. NIS value has a χ^2 distribution with respect to n degrees of freedom. Consider the following:

$$\mathrm{NIS}_k = \left(\mathbf{z}_k - \mathbf{h}\left(\hat{\mathbf{x}}_k^+, k\right)\right)^T \mathbf{S}_k^{-1} \left(\mathbf{z}_k - \mathbf{h}\left(\hat{\mathbf{x}}_k^+, k\right)\right) \le \chi^2, \tag{22}$$

where $\mathbf{S}_k$ is the innovation matrix which is defined as $\mathbf{S}_k = \mathbf{H}_k \mathbf{P}_k^- \mathbf{H}_k^T + \mathbf{R}_k$. The NIS value for valid measurements should be within the threshold of the χ^2 distribution. However, the erroneous measurements are discarded when the NIS value lies outside the threshold boundary. A threshold value that corresponds to a 95% confidence region is used for outlier rejection.

4. Experimental Results

4.1. Experimental Setup. Figure 6 shows the sensor system attached to the mobile robot. The outdoor mobile robot is a Pioneer P3-AT, a commercial outdoor platform of MobileRobots. The wheel encoders attached to the driving motors provide odometry information. SICK LMS-100 was used to detect the curb. The LRF was tilted by 5° toward the ground. A Novatel DGPS system was used to measure the global position. The robot was equipped with a rover antenna, and a base station was located on the roof of a nearby building. The "true" value for the curb and the ground truth were measured using an additional LRF (HOKUYO URG-04lx) attached to the side of the robot.

The experiment was performed in Korea University in Seoul, Korea, as shown in Figure 7. The robot was driven manually along the yellow dashed line from "S" to "G". The curbs along the experimental path are represented as solid green lines. The target environment has semistructured geometry that is composed of paved roads and curbs in most of the region. Furthermore, there are tall buildings and tunnels that degrade the DGPS precision. The travel distance along the experimental path was 775 m, and average speed of the robot was 0.5 m/s.

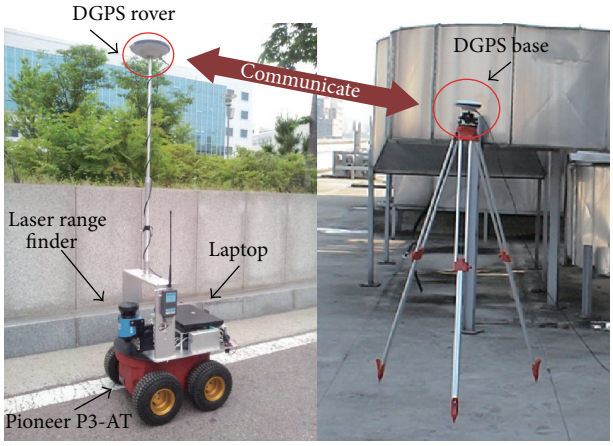


Figure 6: Sensor system attached to mobile robot.

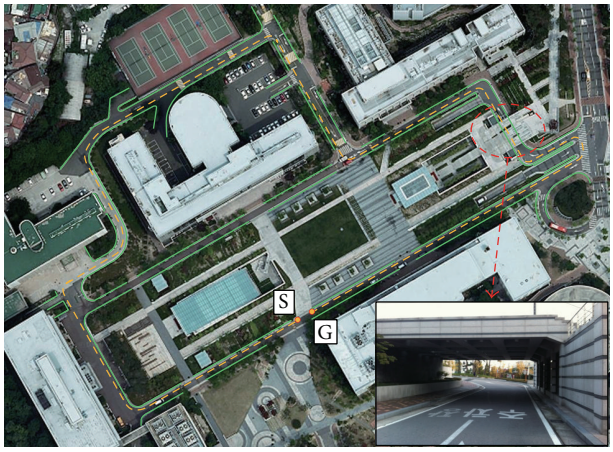


Figure 7: Experimental environment.

4.2. Curb Extraction Results. The curb extraction process in a semistructured road environment is shown in Figure 8. Figure 8(a) shows the road environment where the experiment was performed. The red dotted line represents the scan area of the tilted LRF. Figure 8(b) shows the extracted road surface on the basis of the scanned LRF data in the road environment. When the road surface is extracted, the line segments that satisfy attribute 1 in Section 2.2 are selected as the curb candidates. Figure 8(c) shows the line segments that correspond to the curb candidates. The first and the second candidates for each side of the curbs are represented by blue and green lines, respectively. There are two candidates on each side. Therefore, four pairs of candidates can be extracted as curbs: (R1, L1), (R1, L2), (R2, L1), and (R2, L2). The class prediction results for each pair of curb candidates are listed in Table 1. The pairs of candidates (R1, L1) and (R1, L2) are classified as the curb class. Finally, candidate (R1, L1) is extracted as the curbs as shown in Figure 8(d), which has the smallest Mahalanobis distance with respect to the curb class. If the LRF data that correspond to the vertical surface are detected, the proposed method will perform successfully despite the small and less distinct curbs.

Table 1: Class prediction for pairs of curb candidates.

Candidates	Mahalanobis distance		Class prediction
	Curb class	Noncurb class	
X_1 (R1, L1)	0.0005	163.4922	True Curb
X_2 (R1, L2)	1.9779	202.5195	True Curb
X_3 (R2, L1)	105.6307	4.1180	False Curb
X_4 (R2, L2)	110.2497	3.2296	False Curb

Table 2: Classification results including confusion matrix.

Instance	PCA		KFDA	
Class correct	4338	91.3%	4683	98.6%
Class incorrect	413	8.7%	68	1.4%
Total	**4751**	**100%**	**4751**	**100%**
	Classified as		Classified as	
	True curb	False curb	True curb	False curb
True curb	3524 (97.1%)	106 (2.9%)	3566 (98.2%)	64 (1.8%)
False curb	307 (27.4%)	814 (72.6%)	4 (0.4%)	1117 (99.6%)

The curb extraction was performed while the robot navigates through the experimental path. Figure 9 shows the curb mapping result from the curb extraction. The curb map is shown in the right bottom in Figure 9. The extracted curbs are denoted by the magenta dots. The results indicate that most of the curbs along the experimental path were successfully extracted. In order to demonstrate the robustness of the proposed method, comparison of classification performance between our previous method and the proposed method is summarized in Table 2. The accuracy of the curb extraction with PCA was 91.3% for 4751 scanned datasets. The accuracy was improved to 98.6% with the proposed KFDA. A confusion matrix is presented below. The conventional method and the proposed method show 97.1% and 98.2% of true positive rate, respectively. The curb extraction is performed successfully by the application of both methods. However, the most important factor in the curb extraction is to reduce the false detection rate. False detection is that the noncurb data are misclassified as the curb data. The accuracy of the estimated pose can be decreased, when the false curbs are used for localization. The false detection rate was 27.4% with PCA. The false detection rate was reduced to 0.4% with the proposed KFDA. The result implies that most of the noncurb data were classified as noncurb data correctly.

4.3. Uncertainty Measurement Results. The DGPS measurements are represented by red dots along the experimental path in Figure 10(a). The areas with large DGPS errors are represented by A–F. Figure 10(b) shows the standard deviation of the DGPS measurements. The standard deviation of the DGPS is about 2 m in open area. There were temporal blackouts in areas B, D, and E due to satellite blockage.

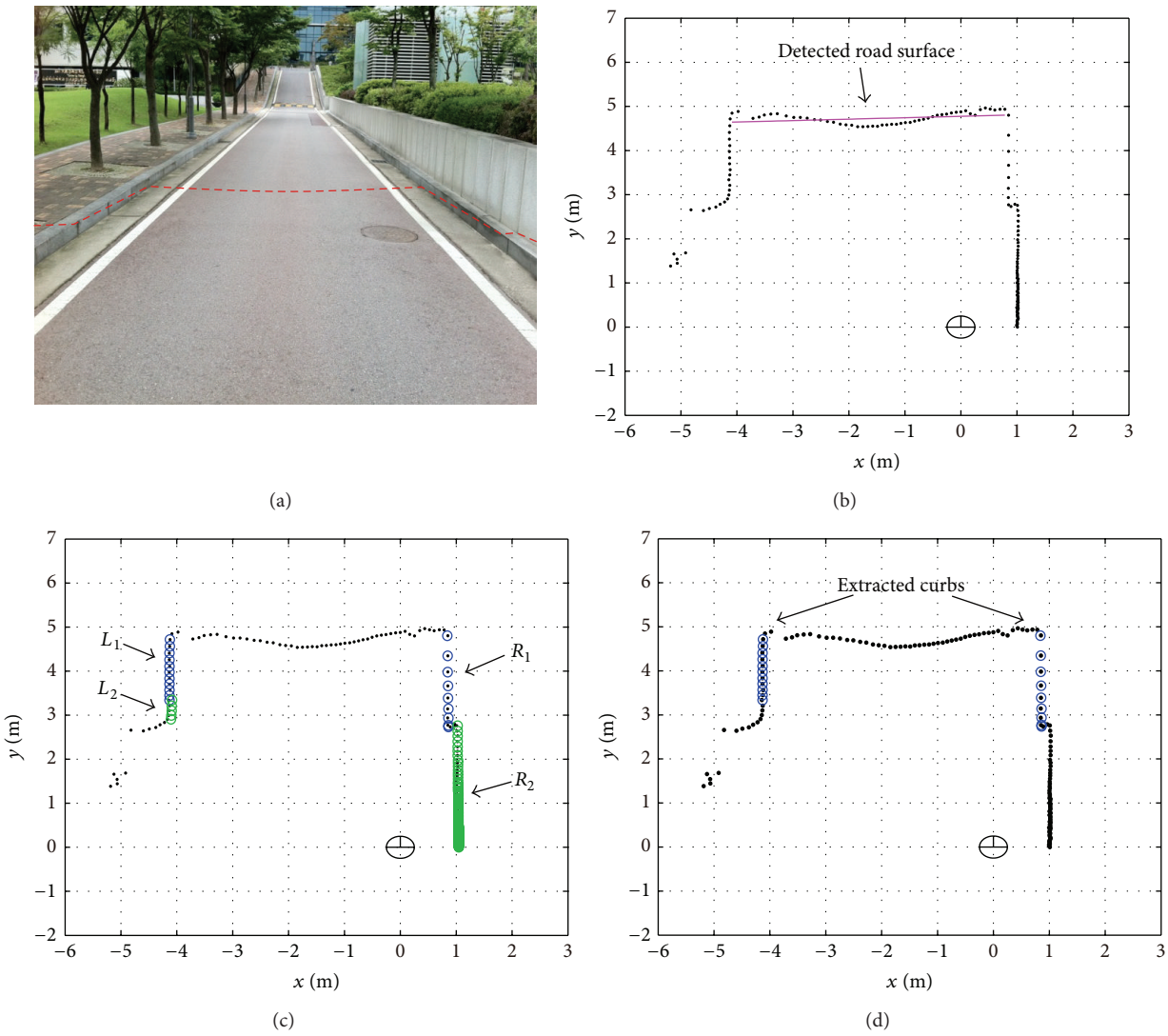


FIGURE 8: Curb extraction results. (a) Road environment. (b) Road surface detection. (c) Curb feature candidates on both sides. (d) Extracted curb.

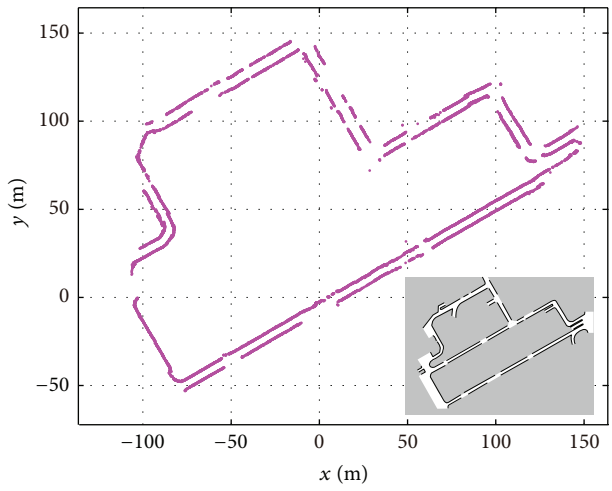


FIGURE 9: Curb mapping result on experimental path.

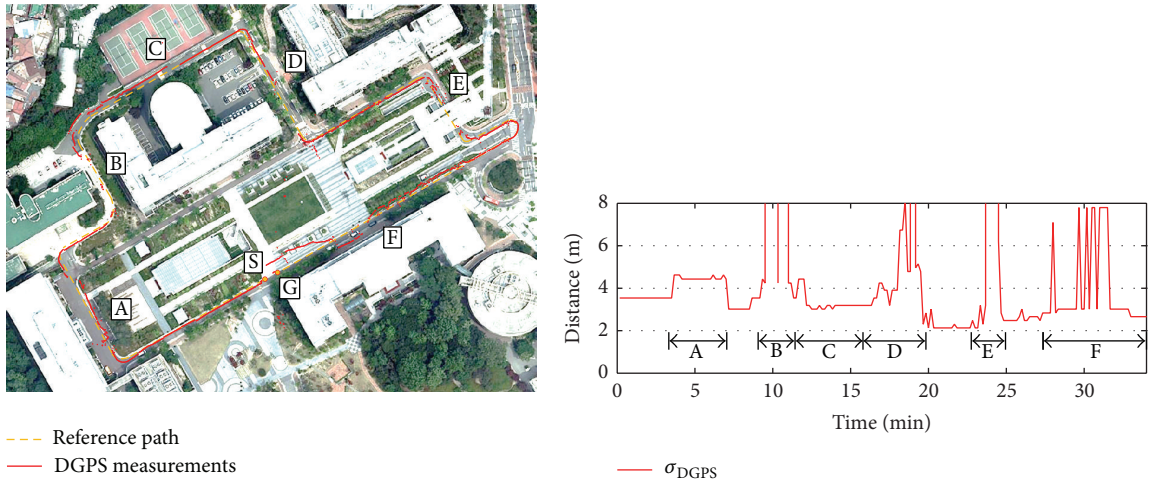


Figure 10: (a) DGPS measurement and (b) DGPS uncertainty measurement (1σ bound).

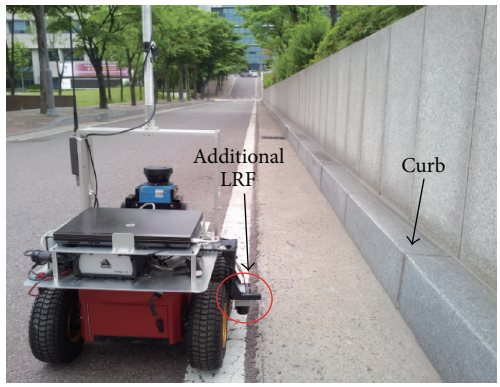


Figure 11: Additional LRF for ground truth of the curbs.

The uncertainty measurements in areas B, D, and E were almost infinite as shown in Figure 10(b). Furthermore, large DGPS errors occurred in areas A, C, and F due to the degraded satellite signals. In particular, measurement errors were 2.1 m on average in area F. The standard deviation that was measured in area F was 6.4 m on average. The result implies that the regional properties of DGPS were obtained by measuring the DGPS uncertainty in real time. However, the precision of the localization only by DGPS can be decreased when the robot navigates near the buildings.

The curb estimation errors are considered in order to define the error covariance. The most accurate result for the quantitative curb uncertainty is obtained by measuring the estimation errors in experimental environments. The estimation errors were measured while the robot navigates through the road with curbs. Ground truth for the curb was provided by an additional LRF that is attached to the side of the robot as shown in Figure 11. Ground truth of the curb was provided by line feature from the additional LRF data. Experiments were conducted in order to measure the estimation errors prior to the localization experiment. The following results show the estimation errors for the range and angle of the extracted curb.

Table 3: Error covariance of extracted curb.

	Range std. σ_r	Angle std. σ_α	Correlation $\sigma_{r\alpha}$
Right curb	0.1620 m	0.0575 rad	0.0035 m·rad
Left curb	0.1614 m	0.0649 rad	−0.0034 m·rad

Figure 12 shows the histogram of the curb estimation errors for each side. The number of the estimation error measurements is approximately 5000. The covariance matrix of the curb was computed from the distributions of estimation errors. The elements of error covariance for each side of the curb are listed in Table 3.

The covariance representation for the extracted curbs is shown in Figure 13. The extracted curbs are represented by blue lines, and the error bounds for the curbs are represented by green dotted lines. The resultant curbs exist within the error bounds with 95% confidence.

4.4. Outdoor Localization Results. The localization results are shown in Figure 14. The proposed method was compared with the conventional framework for outdoor localization that combines DGPS measurement with odometry. The blue dash-dot line represents the localization results corrected only by the DGPS measurements. The estimated paths show some difference with respect to the reference path (yellow dotted line) in area A–F due to the DGPS errors. The magenta line represents the localization results corrected by DGPS and the extracted curb. When the curb information is applied, the results show that the robot position well matches the reference path, even if the DGPS errors are large or a blackout occurs (area A–F). It is clear that the curb information plays a dominant role. The following part presents the localization errors in area A–F, as shown in Figure 14. The localization errors were compared with the conventional framework that combines DGPS measurement with odometry. The ground

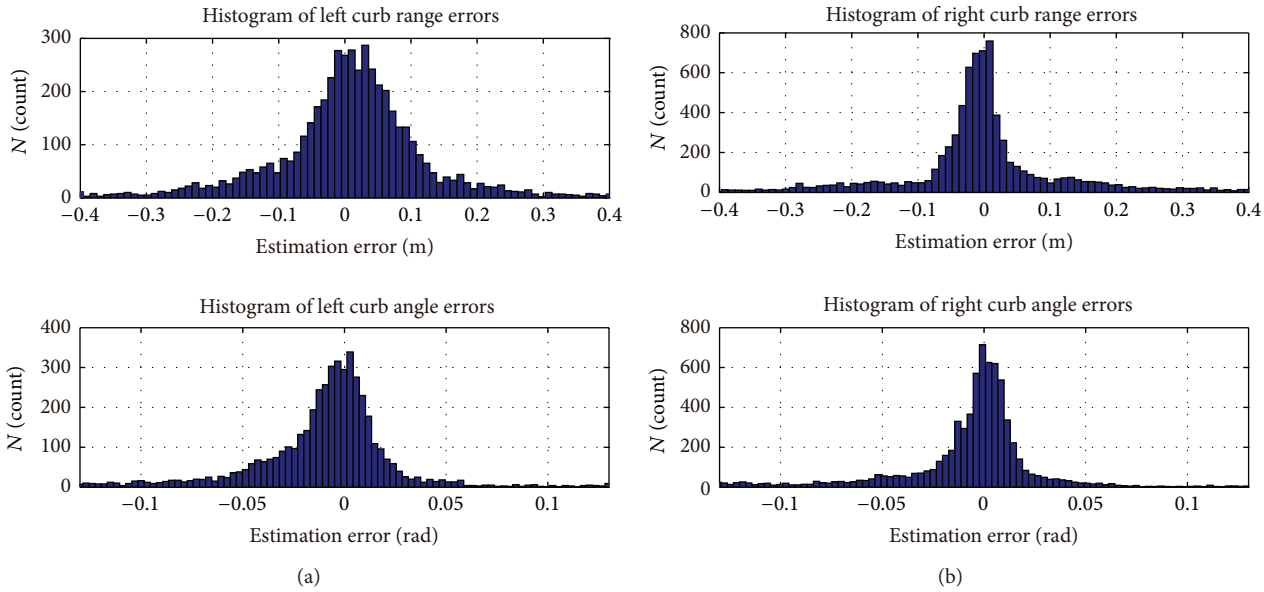


Figure 12: Distribution of estimation errors for (a) left and (b) right curbs.

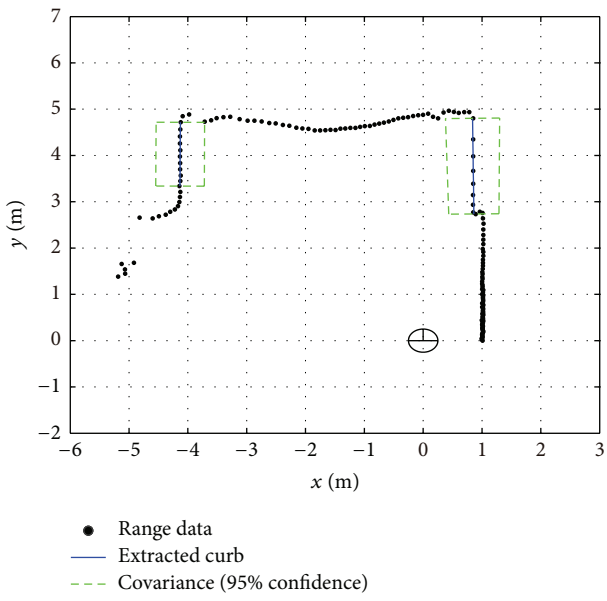


Figure 13: Extracted curbs and covariance representation.

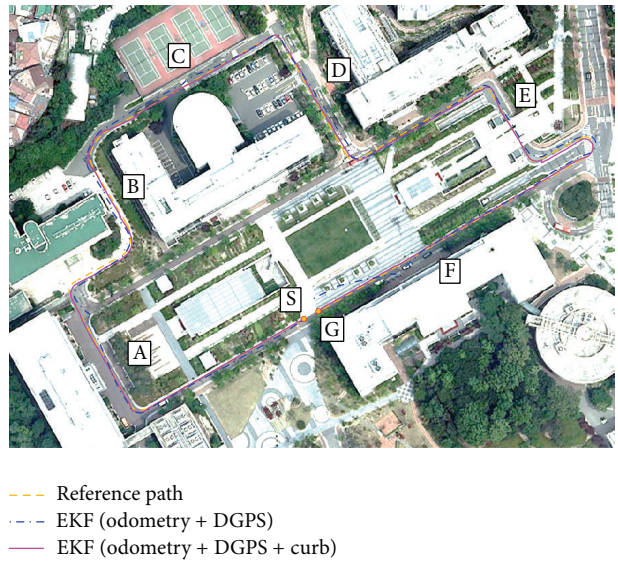


Figure 14: Localization results in semistructured road environment.

truth was computed using the additional LRF attached to the side of the robot.

Figure 15 shows the lateral errors of the localization results. The position errors when using the fusion of the odometry and DGPS are shown in Figure 15(a). The lateral errors in each area were usually greater than 1 m. In particular, the maximum error in area F was 5 m. In contrast, the localization result which was corrected by odometry, DGPS, and curb information shows that the lateral errors were within 0.6 m across the entire area, as shown in Figure 15(b). Therefore, the proposed method shows robust performance in terms of lateral position estimation errors despite the large DGPS errors. The heading errors of the localization results are represented in Figure 16. The result that was corrected using only the DGPS data is shown in Figure 16(a). The heading errors were greater than 1° on average. The variance was larger than 2°. However, when the curb information is used for correction, the heading errors were remarkably decreased. Furthermore, the errors rarely exceed 3° in the entire area. In experiments, the proposed method showed precise and robust performance for the lateral and heading errors, despite the large DGPS errors and a temporal blackout.

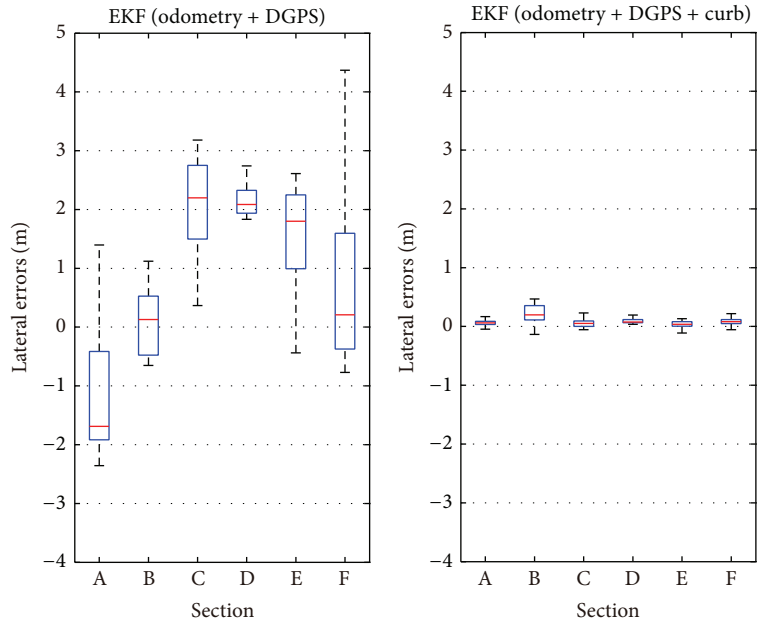


Figure 15: Lateral errors (a) corrected by DGPS and (b) corrected by DGPS and extracted curb.

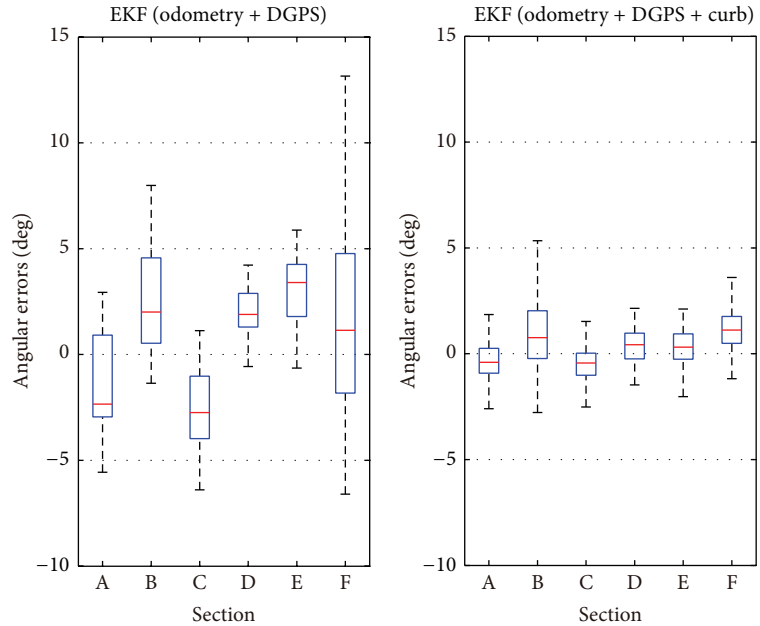


Figure 16: Heading errors (a) corrected by DGPS and (b) corrected by DGPS and extracted curb.

5. Conclusion

This paper presents a localization method for outdoor robots using curb features in semistructured road environments. A reliable curb extraction scheme is proposed to classify the curb candidates as curb and noncurb classes. Most of the curbs in an experimental path are extracted with high accuracy. An EKF-based localization is also proposed to combine the extracted curbs with odometry and DGPS measurements. The uncertainty models of the sensors are defined by experiments to provide a practical solution for localization. From experimental results, the robustness of the proposed method is demonstrated in real road experiments. The curb features can correct significantly the lateral position and heading errors in dense area, where the DGPS signal gets degraded by buildings.

Conflict of Interests

The authors declare that there is no conflict of interests regarding the publication of this paper.

Acknowledgments

This research was supported in part by the MKE (The Ministry of Knowledge Economy), Korea, under the Human Resources Development Program for Convergence Robot Specialists Support Program supervised by the NIPA (National IT Industry Promotion Agency) (NIPA-2013-H1502-13-1001). This work was also supported by the National Research Foundation of Korea (NRF) Grant funded by Korea Government (MEST) (2013-029812).

References

[1] S. Thrun, M. Montemerlo, H. Dahlkamp et al., "Stanley: the robot that won the DARPA Grand Challenge," *Journal of Field Robotics*, vol. 23, no. 9, pp. 661–692, 2006.

[2] M. Buehler, K. Lagnemma, and S. Singh, *The DARPA Urban Challenge: Autonomous Vehicles in City Traffic*, Springer, New York, NY, USA, 2009.

[3] A. Soloviev, "Tight coupling of GPS, laser scanner, and inertial measurements for navigation in urban environments," in *Proceedings of the IEEE/ION Position, Location and Navigation Symposium*, pp. 511–525, Monterey, Calif, USA, May 2008.

[4] S. Panzieri, F. Pascucci, and G. Ulivi, "An outdoor navigation system using GPS and inertial platform," *IEEE/ASME Transactions on Mechatronics*, vol. 7, no. 2, pp. 134–142, 2002.

[5] E.-J. Jung and D.-J. Yi, "Task-oriented navigation algorithms for an outdoor environment with colored borders and obstacles," *Intelligent Service Robotics*, vol. 6, no. 2, pp. 69–77, 2013.

[6] S. Kim, J. Kang, and M. Jin Chung, "Probabilistic voxel mapping using an adaptive confidence measure of stereo matching," *Intelligent Service Robotics*, vol. 6, no. 2, pp. 89–99, 2013.

[7] M. Joerger and B. Pervan, "Range-domain integration of GPS and laser-scanner measurements for outdoor navigation," in *Proceedings of the 19th International Technical Meeting of the Satellite Division of The Institute of Navigation (ION GNSS '06)*, pp. 1115–1123, Fort Worth, Tex, USA, September 2006.

[8] M. Hentschel, O. Wulf, and B. Wagner, "A GPS and laser-based localization for urban and non-urban outdoor environments," in *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 149–154, Nice, France, September 2008.

[9] A. Georgiev and P. K. Allen, "Localization methods for a mobile robot in urban environments," *IEEE Transactions on Robotics*, vol. 20, no. 5, pp. 851–864, 2004.

[10] Y. Morales, T. Tsubouchi, and S. Yuta, "Vehicle localization in outdoor mountainous forested paths and extension of two-dimensional road centerline maps to three-dimensional maps," *Advanced Robotics*, vol. 24, no. 4, pp. 489–513, 2010.

[11] W. S. Wijesoma, K. R. S. Kodagoda, and A. P. Balasuriya, "Road-boundary detection and tracking using ladar sensing," *IEEE Transactions on Robotics and Automation*, vol. 20, no. 3, pp. 456–464, 2004.

[12] M. Jabbour and P. Bonnifait, "Global localization robust to GPS outages using a vertical ladar," in *Proceedings of the 9th International Conference on Control, Automation, Robotics and Vision*, pp. 1–6, December 2006.

[13] P. Bonnifait, M. Jabbour, and V. Cherfaoui, "Autonomous navigation in urban areas using GIS-managed information," *International Journal of Vehicle Autonomous Systems*, vol. 6, no. 5, pp. 83–103, 2008.

[14] Y. Shin, D. Kim, H. Lee, J. Park, and W. Chung, "Autonomous navigation of a surveillance robot in harsh outdoor environments," *Advances in Mechanical Engineering*, vol. 2013, Article ID 837484, 15 pages, 2013.

[15] Y. Shin, C. Jung, and W. Chung, "Drivable road region detection using a single laser range finder for outdoor patrol robots," in *Proceedings of the IEEE Intelligent Vehicles Symposium (IV '10)*, pp. 877–882, San Diego, Calif, USA, June 2010.

[16] D. Kim and W. Chung, "Localization of outdoor mobile robots using road features," in *Proceedings of the 8th International Conference on Ubiquitous Robots and Ambient Intelligence (URAI '11)*, pp. 363–365, Incheon, Republic of Korea, November 2011.

[17] S. Mika, G. Rätsch, J. Weston, B. Schölkopf, and K. Müller, "Fisher discriminant analysis with kernels," *Neural Networks for Signal Processing*, vol. 9, pp. 41–48, 1999.

[18] S. Mika, G. Rätsch, and K. R. Müller, "A mathematical programming approach to the kernel fisher algorithm," *Advances in Neural Information Processing*, vol. 13, pp. 591–597, 2001.

[19] G. Camps-Valls and L. Bruzzone, *Kernel Methods for Remote Sensing Data Analysis*, John Wiley & Sons, New York, NY, USA, 2009.

[20] R. E. Kalman, "A new approach to linear filtering and prediction problems," *Journal of Basic Engineering*, vol. 82, no. 1, pp. 35–45, 1960.

[21] R. E. Kalman and R. S. Bucy, "New results in linear filtering and prediction theory," *Journal of Basic Engineering*, vol. 83, no. 3, pp. 95–108, 1961.

[22] J. J. Leonard and H. F. Durrant-Whyte, *Directed Sonar Sensing for Mobile Robot Navigation*, Kluwer Academic, Dordrecht, The Netherlands, 1992.

[23] B. W. Parkinson and J. J. Spilker, *Global Positioning System: Theory and Applications*, The American Institute of Aeronautics and Astronautics, Washington, DC, USA, 1996.

[24] K. O. Arras and R. Y. Siegwart, "Feature extraction and scene interpretation for map-based navigation and map building," in *12th Mobile Robots Conference*, vol. 3210 of *Proceedings of SPIE*, pp. 42–53, Pittsburgh, Pa, USA, January 1998.

[25] S. T. Pfister, S. I. Roumeliotis, and J. W. Burdick, "Weighted line fitting algorithms for mobile robot map building and efficient data representation," in *Proceedings of the IEEE International Conference on Robotics and Automation*, pp. 1304–1311, September 2003.

Hindawi Publishing Corporation
Mathematical Problems in Engineering
Volume 2014, Article ID 461237, 10 pages
http://dx.doi.org/10.1155/2014/461237

Research Article

An Improved VFF Approach for Robot Path Planning in Unknown and Dynamic Environments

Jianjun Ni,[1,2] Wenbo Wu,[1] Jinrong Shen,[3] and Xinnan Fan[1]

[1] *College of IOT Engineering, Hohai University, Changzhou 213022, China*
[2] *Changzhou Key Laboratory of Sensor Networks and Environmental Sensing, Hohai University, Changzhou 213022, China*
[3] *College of Mechanical and Electrical Engineering, Hohai University, Changzhou 213022, China*

Correspondence should be addressed to Jianjun Ni; njjhhuc@gmail.com

Received 29 December 2013; Revised 1 February 2014; Accepted 5 February 2014; Published 13 March 2014

Academic Editor: Leo Chen

Robot path planning in unknown and dynamic environments is one of the hot topics in the field of robot control. The virtual force field (VFF) is an efficient path planning method for robot. However, there are some shortcomings of the traditional VFF based methods, such as the local minimum problem and the higher computational complexity, in dealing with the dynamic obstacle avoidance. In this paper, an improved VFF approach is proposed for the real-time robot path planning, where the environment is unknown and changing. An area ratio parameter is introduced into the proposed VFF based approach, where the size of the robot and obstacles are considered. Furthermore, a fuzzy control module is added, to deal with the problem of obstacle avoidance in dynamic environments, by adjusting the rotation angle of the robot. Finally, some simulation experiments are carried out to validate and demonstrate the efficiency of the proposed approach.

1. Introduction

Path planning is one of the important issues in the field of robot navigation. The goal of path planning is to find an optimal or suboptimal path from the starting position to the target position, which has been widely used in intelligent transportation system, aerospace, military reconnaissance, family services, underwater exploration, and so forth [1–6]. Various methods have been used to solve the path planning problem. For example, Jaradat et al. [7] proposed a fuzzy-based potential field method for autonomous mobile robot motion planning in dynamic environments including static or moving target and obstacles. Lee et al. [8] presented an approach to a time and energy efficient online complete coverage solution for a mobile robot by using a high-resolution grid map representation to reduce directional constraints on path generation. Antonelli et al. [9] proposed a path following approach based on a fuzzy logic set of rules, which emulates the human driving behavior. Tian et al. [10] used a brain-inspired simultaneously localization and mapping system to construct a spatial cognitive map of an office environment and then a global path is extracted from the built cognitive map and subsequently used by a local planner to instruct the robot to navigate. Many other artificial intelligent methods have been used to deal with the problem of path planning, such as genetic algorithm [11], particle swarm optimization [12], neural network [13], and bioinspired intelligent method [14]. However, these methods have some limitations. For example, the robot will encounter the local minimum problem by the traditional artificial potential field or virtual force field methods [15]; the methods based on grid map require a high completeness of the environmental information [16]; the fuzzy rule base of fuzzy logic methods is often incomplete [9]; the computation of neural network methods is complex, and the traditional neural network methods need a learning process, which cannot meet the requirements of real-time applications sometimes [17, 18].

Aiming at the problems above, some improvements have been proposed recently. Ye et al. [19] presented a virtual obstacles based path planning method for mobile robot in urban road environment, which is based on the benefits of both global and local path planners. Lei and Li [20] proposed a method of behavior-based control for mobile robot path planning in unknown environments using fuzzy logic.

Song and Liu [21] proposed an improved artificial potential field method, where the velocity vector is used to modify the potential field force function, and the fuzzy control method is used to adjust the factors of repulsion potential field in real time. Ni and Yang [22] presented a bioinspired neural network for real-time path planning for multirobot cooperative hunting in unknown environments. Those methods introduced above have some advantages; however, much research on path planning focused on algorithms, and few considered real-world problems, such as the size of obstacles and robots. Furthermore, most of methods in the literature are complex, which are not suitable for real-time applications.

In this paper, an improved virtual force field (VFF) approach is proposed to deal with the path planning problem in unknown and dynamic environments. In the proposed approach, an area ratio parameter is introduced into the traditional VFF based method, considering the size of obstacles and robots in real-world applications. In addition, a fuzzy control module is used to adjust the rotation angle of the robot based on the assessment of threats from environments. By this way, the security of the robotic movement can be guaranteed. The experimental results show that the proposed approach is capable of completing the path planning task efficiently in various situations.

The paper is organized as follows. In Section 2, the problem statement is given and the fundamental of VFF method is introduced. Section 3 presents the proposed path planning approach. Some simulation experiments are conducted, and the results are discussed in Section 4. At last, the conclusions are given in Section 5.

2. Problem Statement and Fundamental of VFF Method

In this paper, the problem of robot path planning in unknown and dynamic environments is studied. The problem is defined as follows. (1) The robot can obtain its location information in real time by some location technologies. (2) The robot has no knowledge about the environment, except for the location of the target. (3) There are a lot of obstacles in the environment, denoted by $\Omega = \{O_i, i = 1, 2, \ldots, m\}$. These obstacles may be static or dynamic. (4) The robot is equipped with some onboard sensors, which are used to detect the obstacles and the target. (5) In this paper, it is assumed that the robot can change the moving direction without delay and move forward at a constant speed. The next location of the robot is

$$\begin{aligned} (x_r)_{t+1} &= (x_r)_t + v_r \, \Delta t \cos (\Phi_r)_t, \\ (y_r)_{t+1} &= (y_r)_t + v_r \, \Delta t \sin (\Phi_r)_t, \end{aligned} \tag{1}$$

where (x_r, y_r), v_r, and Φ_r are the location, the velocity, and the moving direction of the robot, respectively. The problem definition is close to the path planning problem in real world, such as searching for a crashed aircraft positioned by radio transmitters and extinguishing forest fires.

Before the introduction of the proposed approach, the fundamental of the VFF based algorithm is given out. Borenstein and Koren developed VFF method, and the basic idea of the VFF based algorithm is to divide the space of robot into two different computational fields [23, 24]. One computational field is defined as low potential energy field, according to the position of the target relative to the instantaneous position of the robot. This field is concave and minimal in the target position, which is also called the attraction field. The other one is defined as high potential energy field, according to the relative position of the obstacles with respect to the robot position. This field is also called the repulsion field. Based on the attraction and repulsion forces computed from the two different computational fields, the resultant force can be obtained. The resultant force is used to lead the robot. A simple example of path planning based on the VFF method is shown in Figure 1. In this example, there are two obstacles, one target, and one robot. The detailed introduction of the traditional VFF based method can be viewed in [23–26], and a brief introduction of the traditional VFF based method is given out as follows.

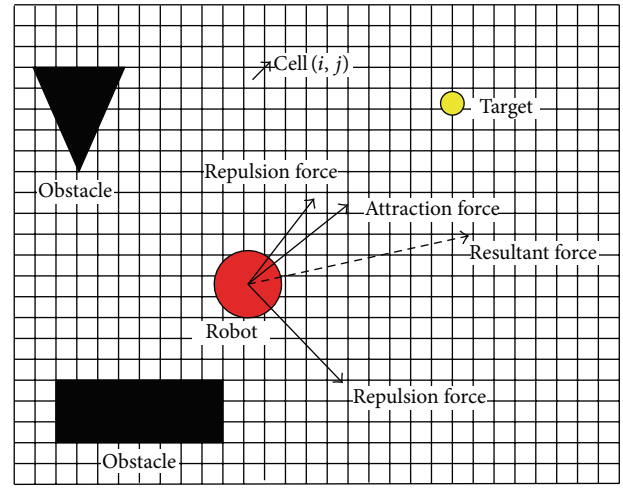


Figure 1: A simple example of path planning based on the VFF method.

(1) The virtual attraction force F_a from the target can be calculated by

$$F_a = F_{ca} \left[\frac{x_t - x_r}{d_t} x' + \frac{y_t - y_r}{d_t} y' \right], \tag{2}$$

where F_{ca} is the attraction force parameter, which is a constant; (x_t, y_t) is the location of the target; (x_r, y_r) is the location of the robot; d_t is the distance between the robot and the target at time t; x' and y' are the unit vectors at x-axis and y-axis, respectively.

(2) The virtual repulsion force F_r from obstacles can be calculated by

$$F_r = \sum_{i,j} \frac{F_{cr} C_{ij}}{d_{ij}^2} \left[\frac{x_i - x_r}{d_{ij}} x' + \frac{y_j - y_r}{d_{ij}} y' \right], \tag{3}$$

where F_{cr} is the repulsion force parameter, which is a constant; C_{ij} is the occupied credibility of the cell (i, j) in the environment by the obstacles; (x_i, y_j) is the center location of the cell (i, j); and d_{ij} is the distance between the robot and the center of the cell (i, j). Then

the resultant force F to the robot in the environment can be calculated by $F = F_a + F_r$; the angle of the vector F is denoted by θ_F, which is used as the moving direction Φ_r for the robot at the next time, namely,

$$(\Phi_r)_{t+1} = \theta_F. \quad (4)$$

3. The Proposed VFF Based Path Planning Approach

As introduced above, the traditional VFF algorithm is a combination of grid based method and the artificial potential field method, which needs to set up the environmental model based on grid map firstly [23, 24]. So there are some limitations in the traditional VFF based method; for example, there is an accumulated error and it needs a large storage space in the grid based method. It is not suitable for real-time applications for the robot navigation in unknown and dynamic environments. In this paper, an improved VFF based path planning approach is proposed. In the proposed approach, the grid method in the traditional VFF based method is simplified, and an area ratio parameter is introduced into the VFF based method to realize the real-time path planning for the robot in unknown environment, which can reduce the computational complexity and solve the local minimum problem in the traditional VFF based method, by adjusting the repulsion forces. Furthermore, a fuzzy control module is used to deal with the obstacle avoidance problem in dynamic environments.

3.1. The Area Ratio Parameter. In this study, an area ratio concept is introduced considering the size of obstacles in the robotic movement. Before the introduction of this area ratio concept, a detection model of the robot is set up. In this paper, the detection range of the robot is represented by a circular area, which is called the perception space. The radius of this circular area is the maximum detection range of the robotic onboard sensors, which is denoted by R, and it is determined by real-world applications. An example of the perception space of the robot is shown in Figure 2. In this example, there are three obstacles detected by the robot.

The area ratio parameter is denoted by δ_i, namely, the area ratio of the ith obstacle and the robot. Because the robot cannot know the shape of the obstacle based on its onboard sensors, it is difficult to determine the area of the obstacle accurately. In this study, the projected area S_i is used to estimate the area of the ith obstacle in the robotic perception space, which can be calculated as follows:

$$S_i = \frac{\alpha_i \left(R^2 - {l_i}^2\right)}{2}, \quad (5)$$

where α_i is the subtended angle of the robot to the ith obstacle (its unit is radians); and l_i is the shortest distance between the robot center and the ith obstacle (see Figure 2). Then the value of the area ratio parameter δ_i can be obtained by

$$\delta_i = \frac{S_i}{S_0}, \quad (6)$$

where S_0 is the projected area of the robot in the working plane; namely, $S_0 = \pi r^2$, where r is the radius of the robot.

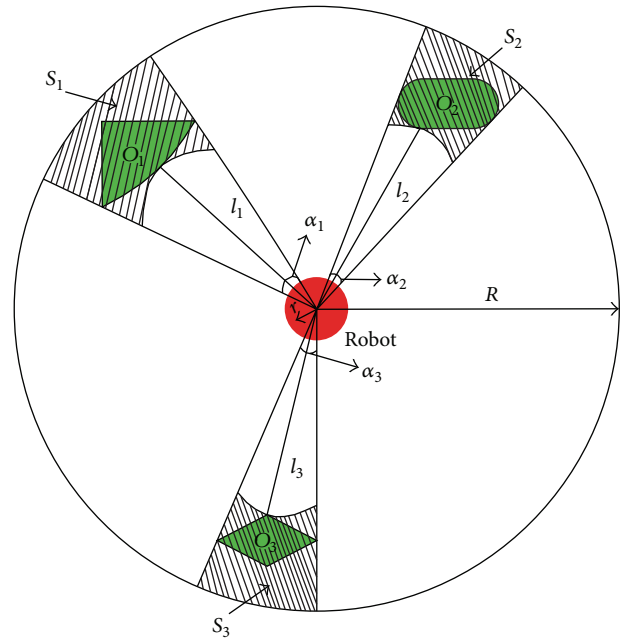


Figure 2: The sketch map of the robotic perception space.

Based on the area ratio parameter δ_i, the adjusted virtual repulsion force to the robot can be calculated by

$$F_r = \sum_{i=1}^{m} \delta_i F_r^i, \quad (7)$$

where m is the number of the obstacles in the robotic perception space, and F_r^i is the ith repulsion force from the ith obstacle, which can be obtained based on a simplified computation from (3); namely

$$F_r^i = \frac{F_{cr}}{d_{o_i}^2}\left[\frac{x_i - x_r}{d_{o_i}}x' + \frac{y_i - y_r}{d_{o_i}}y'\right], \quad (8)$$

where (x_i, y_i) is the center location of the ith obstacle and d_{o_i} is the distance between the center of the ith obstacle and the robot. Because the shape of the obstacle is unknown, the center of the projected area S_i of the ith obstacle is used to estimate the center of this obstacle, namely, d_{o_i} is:

$$d_{o_i} = \frac{l_i + R}{2}. \quad (9)$$

Remark. All the estimations in this study are conservative, to guarantee the safety of the robot during the process of moving to the target.

3.2. The Fuzzy Control Module. In the traditional VFF based path planning method, the obstacles are always assumed to be static. But as we know, there are often many dynamic obstacles in the real-world robotic applications [27, 28], such as the pets and other robots working in the same environment. In this condition, the robot navigated by the traditional VFF method could collide with the dynamic obstacles sometimes. To deal with this problem, some improved methods have been proposed; however, most of these methods for dynamic

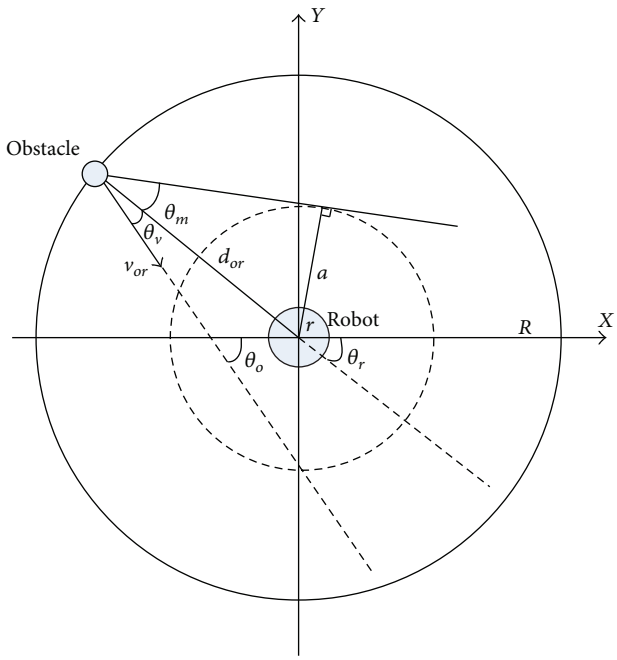


Figure 3: The sketch map of the relationship between the robot and the obstacle.

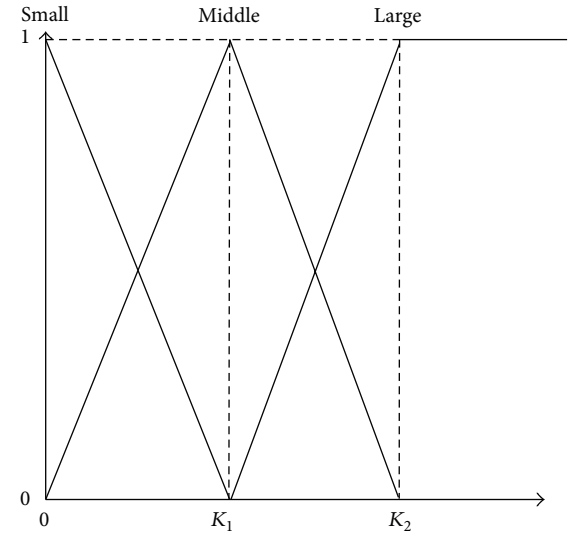


Figure 4: The membership functions of the inputs and output in the fuzzy control module.

environments in the literature are focused on the velocity or acceleration, and few considered the relative position and moving trend of the obstacles and the robot. The real threats of the obstacles have not been assessed, which is very important to reduce the computational complexity of the algorithms [29–31].

In this paper, a fuzzy control module is used in the proposed VFF based method to deal with the dynamic obstacle avoidance. For simplification without losing generality, the dynamic obstacles are assumed to move forward at a constant speed, which can be easily detected. The details of the identification and the computing for the movement parameters of the dynamic obstacles are ignored in this paper. The relative velocity between the robot and the dynamic obstacle is denoted as v_{or} and the angle between the relative velocity vector v_{or} and the line from the robot to the obstacle is denoted as θ_v. The relationship of the robot and the dynamic obstacle is shown in Figure 3.

In Figure 3, the big circle is the detection range of the robot, and its radius is R (it is equal to the maximum detection range of the robotic onboard sensors, see Figure 2); the little circle is the safe range of the robot, and its radius is a, which is a constant less than R. Here, the safe range is just used to judge the risk level of the dynamic obstacles. It does not have a real physical meaning and is different with the definitions in other literature; θ_r is the angle between x-axis and the line from the robot to the obstacle; θ_o is the angle between x-axis and the relative speed vector v_{or}; d_{or} is the distance between the robot and the obstacle; θ_m is the collision angle. From Figure 3, we can see that

$$\theta_v = \theta_o - \theta_r,$$
$$\theta_m = \arcsin \frac{a}{d_{or}}. \tag{10}$$

In this study, the fuzzy control module is based on the VFF based method, which is easily realized. The common methods are adopted in this paper to design this fuzzy control module [18, 32, 33], which are introduced briefly as follows.

(a) The Input and Output of the Fuzzy Control Module. Considering the safety radius of the robot and the effects of the speed of the robot and the dynamic obstacle, two inputs of the fuzzy control module are defined, namely, $\|v_{or} \cos \theta_v\|$ and d_{or}. The two inputs are denoted as λ_1 and λ_2, respectively, which are used to assess the threat of the dynamic obstacle in the detection range. As introduced above, the fuzzy control module is used to adjust the movement direction of the robot based on the threat level of the dynamic obstacle. The output of the fuzzy control module is the rotation angle, denoted by φ. In this study, the sign of the rotation angle φ is decided by the following rules: if the obstacle is located at the left of the robot, the sign of the angle φ is positive; namely, the robot will rotate at a clockwise direction. Otherwise, the robot will rotate at an anticlockwise direction. If the robot and the obstacle are in a line, the robot will rotate an angle φ toward the target.

(b) The Fuzzification and Defuzzification of the Inputs and Output. In this study, the membership function method is used as the fuzzification method of the inputs, and the center of gravity method is used as the defuzzification of the output (see [33, 34]). The membership functions of the inputs and output are shown in Figure 4. The parameters K_1 and K_2 of the membership functions for the inputs and output in Figure 4 are different, which are determined by the experience in real-world applications. In this paper, the values of the parameters K_1 and K_2 are listed in Table 1.

(c) The Fuzzy Control Rules. The fuzzy control rules used in the fuzzy control module can be obtained based on the real-world application experience, which are listed as follows.

Table 1: The values of the parameters in the membership functions.

	The value of K_1	The value of K_2
The input parameter λ_1	v_r	$2v_r$
The input parameter λ_2	$2r$	$4r$
The output parameter φ	$\frac{\pi}{4}$	$\frac{\pi}{2}$

Rule_1: If λ_1 is large and λ_2 is small, then φ is large;

Rule_2: if λ_1 is small and λ_2 is small, then φ is middle;

$\vdots$

Rule_9: if λ_1 is middle and λ_2 is small, then φ is large.

The fuzzy control module will be used when a dynamic obstacle enters the detection range of the robot and $\theta_v < \theta_m$. The inputs are input into the fuzzy control module and then the output can be obtained based on the fuzzy inference of the fuzzy control module. The output is defuzzified and used to adjust the moving angle of the robot obtained by the VFF method introduced in Section 3.1; namely, the moving direction Φ_r for the robot at the next time is adjusted by

$$(\Phi_r)_{t+1} = \theta_F + \varphi. \tag{11}$$

Remark. Although there is much research on the fuzzy logic based path planning methods. It is clear that there are some differences between the literature and the proposed approach in this paper. For example, the fuzzy logic of most of the research is directly used to control the robotic movement, which makes the fuzzy logic base of the fuzzy control module more complex than the proposed approach. Furthermore, the fuzzy control module of the proposed approach is used to adjust the rotation angle obtained by the proposed VFF based method, which realizes the static and dynamic obstacle avoidance automatically and it is different from those approaches based on fuzzy logic directly.

The work flow of the proposed VFF based path planning approach is summarized as follows.

Step 1. Initialize the task; namely, get the location of the target.

Step 2. Perceive the environment by the robotic onboard sensors.

Step 3. Calculate the virtual resultant force for the robot navigation based on the proposed VFF algorithm.

Step 4. Judge whether there are any dynamic obstacles in the perception space of the robot.

Step 5. Adjust the robot moving direction based on the fuzzy control module, if there are any dynamic obstacles found.

Step 6. Navigate the robot movement based on the proposed approach.

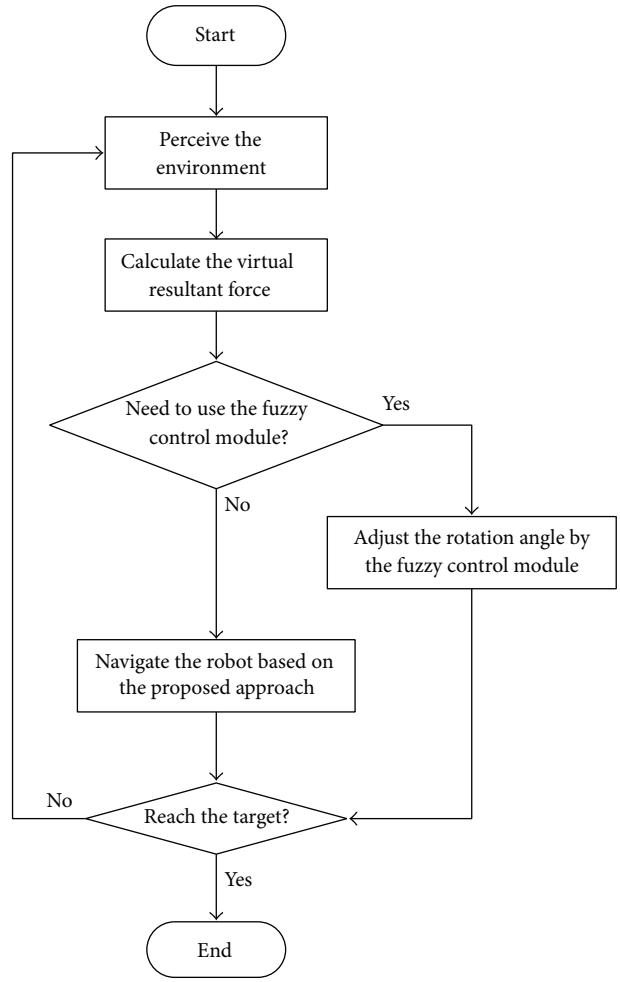


Figure 5: The diagram of the proposed approach.

Step 7. The task is ended if the target is reached; otherwise go to Step 2. The diagram of the proposed approach is shown in Figure 5.

4. Simulation Experiment Studies

To demonstrate the effectiveness of the proposed approach for robot path planning in unknown and dynamic environments, some experiments are conducted that are coded in the Mobotsim software. In these experiments, the environment is set as rectangle. In this paper, two experiments for different situations are conducted, and the parameters in all of the experiments are the same (see Table 2).

4.1. Experiment under the Destination Unreachable State. To test the performance of the proposed approach under the destination unreachable state, this experiment is conducted. In the path planning process based on the traditional VFF method, there could be a destination unreachable state. There are two main conditions that will cause the destination unreachable state. One condition is that the resultant force F could be zero at some time, when the virtual repulsion force

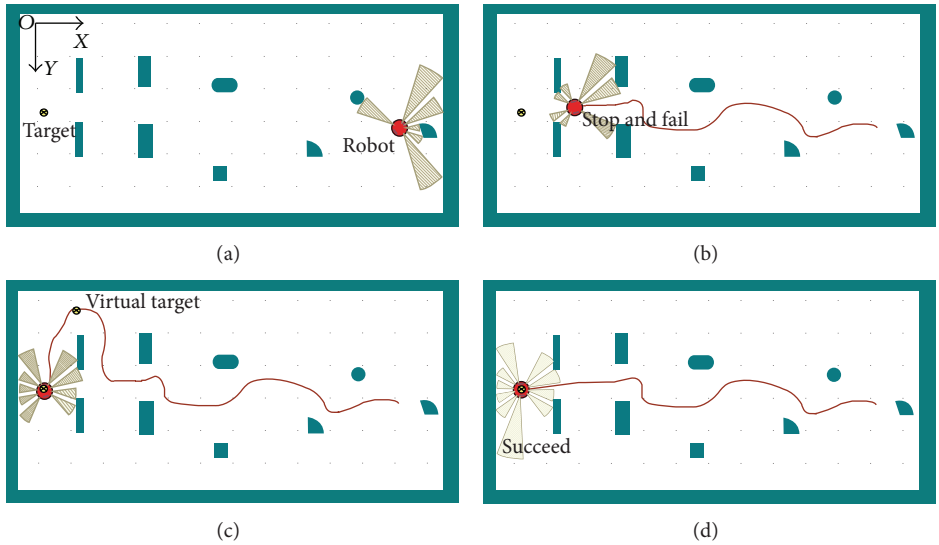


Figure 6: The simulation results in the case of zero force: (a) the initial locations of robot, target, and obstacles, (b) navigated by the T-VFF based method, (c) navigated by the V-VFF based method, and (d) navigated by the P-VFF based method.

Table 2: The values of parameters in the experiments.

Parameters	Value	Remarks
F_{cr}	1	Repulsion constant
F_{ca}	1	Attraction constant
r	0.4 m	Radius of the robot
R	2 m	Detection radius of sensors
v_r	0.5 m/s	Velocity of the robot
a	0.8 m	Safe range of the robot

is equal to the virtual attraction force, but their directions are opposite to each other. The other one is that the obstacle, the target, and the robot are in a line. In this condition, when the robot is near the target, the virtual repulsion force could be bigger than the attraction force [35]. Then the robot will move away from the target and the path planning task will fail.

The unreachable state studied in this paper belongs to the local minimum problem in the path planning process based on the traditional VFF method. To show the advantages of the proposed VFF approach based on the area ratio parameter (P-VFF), it is compared with the traditional VFF based method (T-VFF) and the VFF method based on the virtual target (V-VFF). The basic idea of the virtual target based VFF method is to add a virtual target artificially to deal with the shortcomings of the traditional VFF based methods (see [19, 36]).

The experimental results in the first condition are shown in Figure 6. Figure 6(a) is the initial locations of obstacles, target, and robot. The initial locations of the robot and the target are (17.50, 12.32) and (6.04, 12.06), respectively. Figures 6(b) to 6(d) are the results of path planning for the robot based on T-VFF, V-VFF, and P-VFF methods, respectively. The experimental results in the second condition are shown

Table 3: The performance comparison among the three methods in the case of zero force.

The navigation approach	Required time to the target	Average rotation angle of each step
The T-VFF based method	/	/
The V-VFF based method	15.3 s	11.2°
The P-VFF based method	12.4 s	9.7°

Table 4: The performance comparison among the three methods where robot, target and obstacle are in a line.

The navigation approach	Required time to the target	Average rotation angle of each step
The T-VFF based method	/	/
The V-VFF based method	14.5 s	7.6°
The P-VFF based method	12.5 s	6.9°

in Figure 7. In the experiment of the second condition, the initial locations of the robot and the target are the same as the experiment at the first condition, except that the obstacles in the environment are different. The performance comparison results among the three methods in the two conditions are listed in Tables 3 and 4.

The results in Figures 6 and 7 show that the path planning tasks failed based on the T-VFF method in the two conditions (see Figures 6(b) and 7(b), where the robot will stop at the 9.3th second and the 9.7th second, respectively, before reaching the target). The results in Figures 6(c) and 7(c) show that the robot can complete the path planning task by the V-VFF method in the two conditions. However, the location of the virtual target in the V-VFF method is difficult to be determined, which cannot ensure smoothness of the robotic path to the target. The average rotation angle at each step of the robot navigated by the proposed approach is less than that

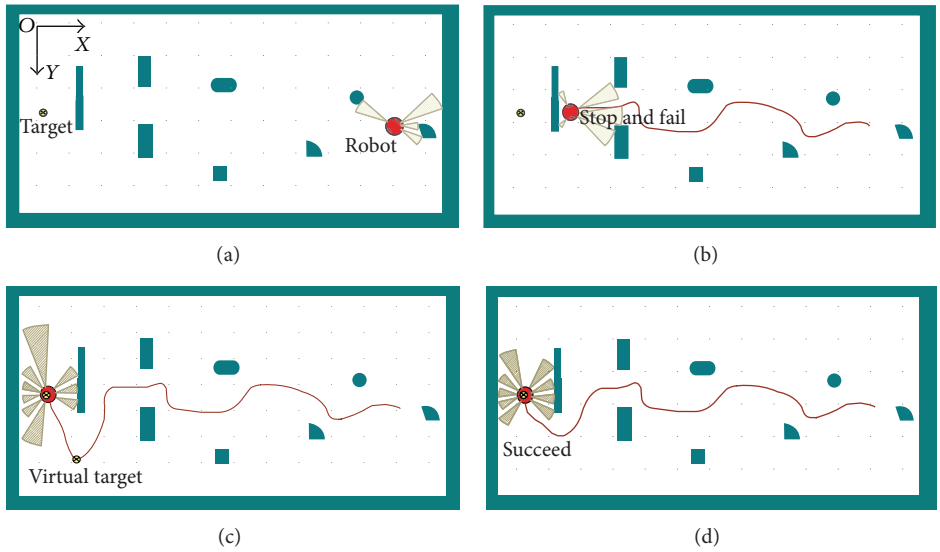


Figure 7: The simulation results where robot, target, and obstacle are in a line: (a) the initial locations of robot, target, and obstacles, (b) navigated by the T-VFF based method, (c) navigated by the V-VFF based method, and (d) navigated by the P-VFF based method.

of the robot navigated by the V-VFF method, which can reduce the energy consumption of the robot (see Tables 3 and 4).

Remark. Of course the V-VFF based method could be better tuned to complete the path planning task more efficiently than the proposed approach, but it needs more experience and time, which are both scarce in the real-world applications of path planning.

The results in the experiments show that the proposed approach can deal with the problems of the path planning task in the two conditions of the destination unreachable state. The trajectory of the robot navigated by the proposed approach is smooth; namely, the energy consumption is less, which is very important for some real-world applications.

4.2. Experiment under Dynamic Environment. To further test the performance of the proposed approach in the dynamic environment, this experiment is conducted. In this experiment, there are two identical robots, which have different targets. One robot is treated as the dynamic obstacle, denoted by O_0. The targets for the dynamic obstacle O_0 and the robot R_0 are denoted as M_O and M_R, respectively. The dynamic obstacle is navigated by the traditional VFF method. In the dynamic environment, it is difficult to determine the location of the virtual target if the robot is navigated by the virtual target based VFF method. So in this experiment, the proposed approach is only compared with the traditional VFF based method (T-VFF); namely, the robot will be navigated by the traditional VFF method firstly and then navigated by the proposed approach (P-VFF), to show the advantages of the proposed approach. In this experiment, two conditions are considered, the first condition is that the robot will move toward the dynamic obstacle, and the second condition is that a faster dynamic obstacle will track and hit the robot.

In the first condition, the initial locations of the robot R_0 and the dynamic obstacle O_0 are (8.56, 6.69) and (1.23, 6.54), respectively. The locations of the targets for the robot R_0 and the dynamic obstacle O_0 are (1.45, 4.84) and (7.78, 4.14), respectively (see Figure 8(a)). In this experiment, the velocity of the dynamic obstacle is the same as that of the robot. The experimental results are shown in Figure 8.

The results in Figure 8 show that the robot and the dynamic obstacle will have the risk to collide, when they move 6.6 seconds in opposite direction face to face. At this moment, the distance between the robot and the dynamic is 0.8 m; the angle $\theta_v = 27^\circ$; and the angle $\theta_m = 36^\circ$ (see Figure 8(b)). The robot based on the traditional VFF method will collide with the dynamic obstacle (see Figure 8(c)), while the robot based on the proposed approach can avoid the obstacle efficiently and reach the target successfully (see Figure 8(d)).

In the second condition, the initial locations of the robot R_0 and the dynamic obstacle O_0 are (2.00, 5.24) and (1.12, 5.65), respectively. The locations of the targets for the robot R_0 and the dynamic obstacle O_0 are (7.20, 4.60) and (8.31, 3.94), respectively (see Figure 9(a)). In this experiment, the velocity of the dynamic obstacle is twice as quick as the robot. The experimental results are shown in Figure 9.

The results in Figure 9 show that the robot based on the traditional VFF method cannot avoid the dynamic obstacle with higher speed, although the robot can detect the dynamic obstacle and adjust its moving direction (see Figure 9(c)). However, the robot based on the proposed approach can avoid the dynamic obstacle timely by adjusting its moving direction with the fuzzy rules (see Figures 9(b) and 9(d)).

The results in these experiments show that the robot will collide with the dynamic obstacle sometimes if it is navigated by the traditional VFF method (see Figures 8(c) and 9(c)). The reason is that the repulsion force in the VFF based method will be very complex and change quickly when

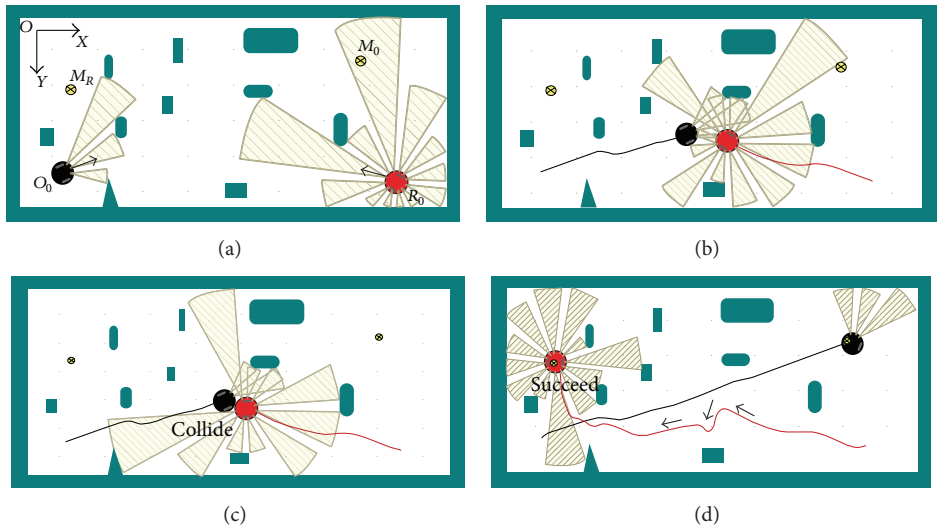


FIGURE 8: The simulation results of the dynamic obstacle avoidance in the first condition: (a) the initial locations of robot, target, and obstacles, (b) the status that the robot has the risk of collision with the dynamic obstacle, (c) navigated by the T-VFF based method, and (d) navigated by the P-VFF based method.

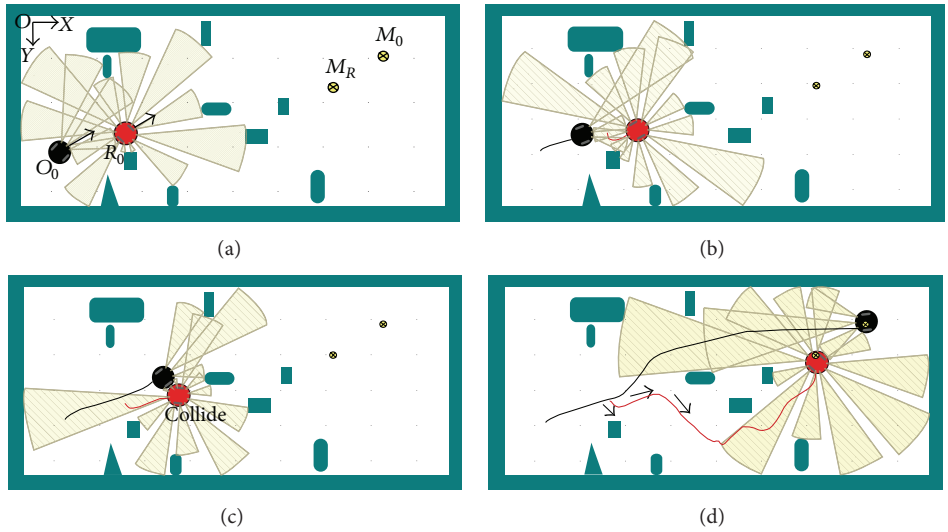


FIGURE 9: The simulation results of the dynamic obstacle avoidance in the second condition: (a) the initial locations of robot, target, and obstacles, (b) the status that the robot has the risk of collision with the dynamic obstacle, (c) navigated by the T-VFF based method, and (d) navigated by the P-VFF based method.

the obstacle is moving. In the proposed approach, a fuzzy control module is used to adjust the moving direction of the robot, so the robot can reach the target safely, without collision with the dynamic obstacle (see Figures 8(d) and 9(d)).

5. Conclusions

The path planning problem for robots in unknown and dynamic environments is investigated. In order to deal with this problem, a novel area ratio parameter based VFF method is proposed. The proposed approach can deal with the destination unreachable state in the path planning process based on the traditional VFF method, by the area ratio parameter considering the size of the obstacles and robots. Furthermore, a fuzzy control module is proposed to deal with the path planning problem in dynamic environments. The path generated by the proposed approach is smooth, which can reduce the energy consumption of the robot. In addition, the robot navigated by the proposed approach can reach the target safely in the dynamic environments. The feasibility and efficiency of the proposed approach are discussed and illustrated through simulation experimental studies. The results show that the proposed approach can complete the path planning task in various situations efficiently, without complex computing or large fuzzy rule base. The proposed approach is applicable to the navigation of the robot in various unknown environments, such as the forest fires extinguishing and underwater navigation [37, 38].

In future work, the real robot experiments should be conducted to test the performance of the proposed approach for path planning in real-world applications considering the physical constraints of the robot, and the parameters in the proposed approach should be optimized to further improve the ability for the robot path planning in complex environments. Some new bioinspired learning method may be considered in the path planning method for robot, which is the development trend in the robot control field [22, 39].

Conflict of Interests

The authors declare that there is no conflict of interests regarding the publication of this paper.

Acknowledgments

This work was supported by the National Natural Science Foundation of China (61203365), the Jiangsu Province Natural Science Foundation (BK2012149), the Open Fund of Changzhou Key Laboratory of Sensor Networks and Environmental Sensing (CZSN201102), the Fundamental Research Funds for the Central Universities (2011B04614), and the Science and Technology Commission of Shanghai Municipality (12595810200).

References

[1] T.-K. Wang, Q. Dang, and P.-Y. Pan, "Path planning approach in unknown environment," *International Journal of Automation and Computing*, vol. 7, no. 3, pp. 310–316, 2010.

[2] J. Ni and S. X. Yang, "A fuzzy-logic based chaos GA for cooperative foraging of multi-robots in unknown environments," *International Journal of Robotics and Automation*, vol. 27, no. 1, pp. 15–30, 2012.

[3] E. Galceran and M. Carreras, "A survey on coverage path planning for robotics," *Robotics and Autonomous Systems*, vol. 61, no. 12, pp. 1258–1276, 2013.

[4] H. Qu, K. Xing, and T. Alexander, "An improved genetic algorithm with co-evolutionary strategy for global path planning of multiple mobile robots," *Neurocomputing*, vol. 120, pp. 509–517, 2013.

[5] M.-C. Tsou and C.-K. Hsueh, "The study of ship collision avoidance route planning by ant colony algorithm," *Journal of Marine Science and Technology*, vol. 18, no. 5, pp. 746–756, 2010.

[6] D. Zhu, H. Huang, and S. Yang, "Dynamic task assignment and path planning of multi-AUV system based on an improved self-organizing map and velocity synthesis method in three-dimensional underwater workspace," *IEEE Transactions on Cybernetics*, vol. 43, no. 2, pp. 504–514, 2013.

[7] M. A. K. Jaradat, M. H. Garibeh, and E. A. Feilat, "Autonomous mobile robot dynamic motion planning using hybrid fuzzy potential field," *Soft Computing*, vol. 16, no. 1, pp. 153–164, 2012.

[8] T.-K. Lee, S.-H. Baek, Y.-H. Choi, and S.-Y. Oh, "Smooth coverage path planning and control of mobile robots based on high-resolution grid map representation," *Robotics and Autonomous Systems*, vol. 59, no. 10, pp. 801–812, 2011.

[9] G. Antonelli, S. Chiaverini, and G. Fusco, "A fuzzy-logic-based approach for mobile robot path tracking," *IEEE Transactions on Fuzzy Systems*, vol. 15, no. 2, pp. 211–221, 2007.

[10] B. Tian, V. A. Shim, M. Yuan, C. Srinivasan, H. Tang, and H. Li, "RGB-D based cognitive map building and navigation," in *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS '13)*, pp. 1562–1567, Tokyo, Japan, November 2013.

[11] V. Roberge, M. Tarbouchi, and G. Labonte, "Comparison of parallel genetic algorithm and particle swarm optimization for real-time UAV path planning," *IEEE Transactions on Industrial Informatics*, vol. 9, no. 1, pp. 132–141, 2013.

[12] Y. Xue and H. Liu, "Optimal path planning in complex indoor environment based on improved PSO," *Journal of Computational Information Systems*, vol. 7, no. 6, pp. 2158–2165, 2011.

[13] M. P. S. Kumar and S. Rajasekaran, "A neural network based path planning algorithm for extinguishing forest fires," *International Journal of Computer Science Issues*, vol. 9, no. 2, pp. 563–568, 2012.

[14] J. A. Villacorta-Atienza, M. G. Velarde, and V. A. Makarov, "Compact internal representation of dynamic situations: neural network implementing the causality principle," *Biological Cybernetics*, vol. 103, no. 4, pp. 285–297, 2010.

[15] S. Saravanakumar and T. Asokan, "Multipoint potential field method for path planning of autonomous underwater vehicles in 3D space," *Intelligent Service Robotics*, vol. 6, no. 4, pp. 211–224, 2013.

[16] L. Murphy and P. Newman, "Risky planning on probabilistic costmaps for path planning in outdoor environments," *IEEE Transactions on Robotics*, vol. 29, no. 2, pp. 445–457, 2013.

[17] R. Araújo, "Prune-able fuzzy ART neural architecture for robot map learning and navigation in dynamic environments," *IEEE Transactions on Neural Networks*, vol. 17, no. 5, pp. 1235–1249, 2006.

[18] D. Liu, L. Wang, and K. Tan, *Design and Control of Intelligent Robotic Systems*, vol. 177 of *Studies in Computational Intelligence*, Springer, Berlin, Germany, 2009.

[19] W. Ye, C. Wang, M. Yang, and B. Wang, "Virtual obstacles based path planning for mobile robots," *Robot*, vol. 33, no. 3, pp. 273–286, 2011.

[20] B. Lei and W. Li, "A fuzzy behaviours fusion algorithm for mobile robot real-time path planning in unknown environment," in *Proceedings of the IEEE International Conference on Integration Technology (ICIT '07)*, pp. 173–178, Shenzhen, China, March 2007.

[21] Q. Song and L. Liu, "Mobile robot path planning based on dynamic fuzzy artificial potential field method," *International Journal of Hybrid Information Technology*, vol. 5, no. 4, pp. 85–94, 2012.

[22] J. Ni and S. X. Yang, "Bioinspired neural network for real-time cooperative hunting by multirobots in unknown environments," *IEEE Transactions on Neural Networks*, vol. 22, no. 12, pp. 2062–2077, 2011.

[23] S. Sarkar, S. Reynolds, and E. Hall, "Virtual force field based obstacle avoidance and agent based intelligent mobile robot," in *25th Intelligent Robots and Computer Vision: Algorithms, Techniques, and Active Vision*, vol. 6764 of *Proceedings of SPIE*, p. 12, Boston, Mass, USA, September 2007.

[24] J. Borenstein and Y. Koren, "Real-time obstacle avoidance for fast mobile robots," *IEEE Transactions on Systems, Man and Cybernetics*, vol. 19, no. 5, pp. 1179–1187, 1989.

[25] J. Carlier and T. Murakami, "Virtual force field based of force-feedback of road condition for driving assistant design in electric vehicle," *International Journal of Automation Technology*, vol. 5, no. 6, pp. 908–915, 2011.

[26] Y. Jin and B. Wang, "Virtual force field integrated with human avoiding obstacles strategies for robot," *Applied Mechanics and Materials*, vol. 16-19, pp. 1071–1076, 2009.

[27] Z. Wu and L. Feng, "Obstacle prediction-based dynamic path planning for a mobile robot," *International Journal of Advancements in Computing Technology*, vol. 4, no. 3, pp. 118–124, 2012.

[28] Y. Lu, F. Qiu, J. Xin, and W. Shang, "Dynamic obstacle avoidance for path planning and control on intelligent vehicle based on the risk of collision," *WSEAS Transactions on Systems*, vol. 12, no. 3, pp. 154–164, 2013.

[29] C. Liu and J. Yang, "Mobile robot path planning based on potential field method in dynamic environments," *Journal of Computational Information Systems*, vol. 6, no. 13, pp. 4435–4444, 2010.

[30] L. Yin, Y. Yin, and C.-J. Lin, "A new potential field method for mobile robot path planning in the dynamic environments," *Asian Journal of Control*, vol. 11, no. 2, pp. 214–225, 2009.

[31] X. Li, R. Wei, and Z. Wang, "Three-dimension path planning for UAV using improved A* algorithm in complicated threat environment," *High Technology Letters*, vol. 17, no. 1, pp. 13–18, 2011.

[32] F.-H. Hsiao, C.-W. Chen, Y.-W. Liang, S.-D. Xu, and W.-L. Chiang, "T-S fuzzy controllers for nonlinear interconnected systems with multiple time delays," *IEEE Transactions on Circuits and Systems I*, vol. 52, no. 9, pp. 1883–1893, 2005.

[33] J. Liu, *Intelligent Controls*, Publishing House of Electronics Industry, Beijing, China, 2005.

[34] C. Ye, N. H. C. Yung, and D. Wang, "A fuzzy controller with supervised learning assisted reinforcement learning algorithm for obstacle avoidance," *IEEE Transactions on Systems, Man, and Cybernetics B*, vol. 33, no. 1, pp. 17–27, 2003.

[35] S. S. Ge and Y. J. Cui, "New potential functions for mobile robot path planning," *IEEE Transactions on Robotics and Automation*, vol. 16, no. 5, pp. 615–620, 2000.

[36] L. Sun, R. Lin, W. Wang, and Z. Du, "Mobile robot real-time path planning based on virtual targets method," in *Proceedings of the 3rd International Conference on Measuring Technology and Mechatronics Automation (ICMTMA '11)*, vol. 2, pp. 568–572, Shanghai, China, January 2011.

[37] Y. Tian, M. Yang, X. Qi, and Y. Yang, "Multi-robot task allocation for fire-disaster response based on reinforcement learning," in *Proceedings of the International Conference on Machine Learning and Cybernetics*, pp. 2312–2317, Baoding, China, July 2009.

[38] T. Maki, T. Ura, and T. Sakamaki, "AUV navigation around jacket structures II: map based path-planning and guidance," *Journal of Marine Science and Technology*, vol. 17, no. 4, pp. 523–531, 2012.

[39] M. J. Milford, *Robot Navigation from Nature: Simultaneous Localisation, Mapping, and Path Planning Based on Hippocampal Models*, vol. 41 of *Springer Tracts in Advanced Robotics*, Springer, Berlin, Germany, 2008.